UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR

Universidad Autónoma
de Madrid

Master in Deep Learning for
Audio and Video Signal Processing

MASTER THESIS

APPLICATION FOR THE DEMONSTRATION OF THE AUTOMATIC REGISTRATION OF TRANSITED SPACES FOR CONTACT TRACING OF INFECTIOUS DISEASES USING VIDEO SIGNALS FROM LIFE-LOGGING CAMERAS.

Daniel De Alcalá Valcárcel
Advisor: Marcos Escudero Viñolo
Lecturer: Jesús Bescós Cano

June 2021

APPLICATION FOR THE DEMONSTRATION OF THE AUTOMATIC REGISTRATION OF TRANSITED SPACES FOR CONTACT TRACING OF INFECTIOUS DISEASES USING VIDEO SIGNALS FROM LIFE-LOGGING CAMERAS.

Daniel De Alcalá Valcárcel
Advisor: Marcos Escudero Viñolo
Lecturer: Jesús Bescós Cano

Dpto. Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid

June 2021

This work has been partially supported by the Spanish Government through its TEC2017-88169-R MobiNetVideo project.

Resumen

En los últimos años, tener un registro de los lugares visitados por una persona es una herramienta muy útil para diversas tareas. Puede utilizarse para rastrear contactos de COVID-19, un tema muy relevante actualmente, o para modelar rutinas de una persona y enseñar a una máquina. El trabajo previo en esto es escaso, por lo que aún no se han logrado buenos resultados. *Ego-topo* fue el primer trabajo que alcanzó un resultado prometedor. El propósito de este trabajo es extender su método, mejorar los resultados y finalmente evaluar su uso en una aplicación real.

En primer lugar, se ha hecho un estudio de los sistemas en los principales campos que han llevado hasta el actual *Ego-topo*. A continuación, se profundizó en los detalles teóricos y algorítmicos del sistema *Ego-topo*, para así entender sus debilidades y fortalezas. Después se modificaron diversos puntos del sistema para lograr funcionalidad añadida, entre ellos habilitar el uso de vídeos externos y la construcción de grafos combinados entre distintos usuarios. Estos grafos combinados son los que permitieron desarrollar una aplicación para detectar automáticamente posibles contactos entre dos usuarios en un entorno doméstico, con el objetivo de un rastreo de enfermedades infecciosas. Antes del desarrollo de la aplicación, se mejoraron algunas debilidades del sistema base que perjudicaban su rendimiento. Estos cambios fueron evaluados tanto subjetivamente, como de forma objetiva con la creación de unas anotaciones de región de manera semi-automática. Finalmente, con estas mejoras se realizó la interfaz gráfica de la aplicación.

Estos son unos primeros pasos en este tema, lo que abre un amplio campo de investigación y aplicaciones. Trabajos futuros pueden transferir este rastreo a entornos de mayor escala, como por ejemplo un aeropuerto, y pueden obtener mejores resultados aprovechándose de otra información como el sonido en distintas áreas.

Palabras clave

Reconocedor, lugar, vídeo, egocéntrico, primera persona, entorno doméstico, interior, EpicKitchens, Ego-topo, grafo, estructurado, Deep Learning, Machine Learning, redes neuronales, automático, COVID-19, rastreador, detector, contactos, enfermedad, contagios.

Abstract

In recent years having a register of the places that a person has visited is a very powerful tool for several tasks. It can even be used for tracing COVID-19 contacts, a ubiquitous topic in these times, or to model routines of a person to teach a machine. Previous work on this is scarce so they do not achieve accurate results. *Ego-topo* was the first work to reach a hopeful performance. The purpose of this thesis is to extend their approach, improve the results, and finally deploy it to a real application.

In the first place, a study has been made of the systems that have led to the current *Ego-topo*. Next, the theoretical and algorithmic details of the *Ego-topo* system were delve into, with the aim of understanding its strengths and weaknesses. Later, several points of the system were modified in order to achieve added functionality, including enabling the use of external videos and the construction of combined graphs between different users. This combined graphs allowed the development of an application to detect contacts between two users in a domestic environment, with the aim of infectious diseases tracing. Before the application development, some weaknesses of the base system that harm its results were improved. The new changes were evaluated first subjectively and then objectively with the creation of semi-automatic region annotations. Finally, with these improvements, the development of the application's graphical interface was carried out.

These are initial steps in this topic which provide a wide field of research and applications. Future works could transfer this illness tracking to bigger scale environments, for example an airport, and they may reach better results benefiting from other features like the sound of different areas.

Keywords

Recognizer, place, video, egocentric, first person, domestic environment, indoor, Epic-Kitchens, Ego-topo, graph, structured, Deep Learning, Machine Learning, neural netowrks, automatic, COVID-19, tracker, detector, contacts, disease, contagious.

Acknoledgements

I would like to thank in first place to my family and friends who has supported me on every step of my life. I would like to express my special thanks of gratitude to my teachers, particularly to Marcos and Alejandro, for the patience, and the implication. But also to all those who gave me the opportunity to do this wonderful project, which helped me in doing a lot of research and to learn about so many new things. I am really thankful. Eternally grateful to Charlie from HB for your art, you are eternal.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The motivation of this work is to make our contribution in the field of automatic place registration. We consider this is a very novel but crucial line of investigation helpful for several tasks such as tracking COVID-19 contacts. Early detection of contacts in the field of infectious diseases could save lives, and COVID-19 taught us our weakness in this regard.

We have selected the state-of-the-art *Ego-topo* system which can create a structured representation of the places visited by an user. After few modifications this approach can be used for computing a larger graph with the information needed to know if two users have passed by the same places. The method is fed with videos captured by an egocentric camera, so it is perfect for a life-logging scenario, a growing field with a lot of possible applications, where the improvement of the place detection could open up a world of possibilities. Even though *Ego-topo* is currently the best system, the possible improvements are abundant, and every advance on the approach may imply paramount steps.

Being part of the growth of this topic and being helpful for promising future research is the most important motivation. There are other impulses, such as investigating itself and learning a bit more about neural networks, algorithms, programming, etc. Also the possibility of creating something that could be useful for the society is a very dominant incentive, and to see with my own eyes the result of the efforts in something usable. The last reason is to improve something enough so it can be published, and start a publishing career that would motivate me a lot.

1.2 Objectives

1. Leverage current state-of-the-art for creating a graph of the places or locations in a selected video, with the nodes representing specific locations (e.g., a sink or a table) and the edges representing spatiotemporal relations between them (e.g., vicinity).

2. Design and develop methods to accumulate video statistics arranged by the graph structure, e.g., the number of visits to a specific place inside a video or the time expended at a given place.

3. Design and develop methods to link locations captured at different videos e.g., the same sink captured at different days, and improve the baseline performance.

4. Design and develop methods to accumulate statistics from different videos of the same scene arranged by the graph structure.

5. Explore potential applications of these created graphs.

1.3 Structure

This report has the following chapters

- chapter 1 Introduction.

  Brief description of the reasons why this work is done and the objectives.

- chapter 2 State of the art.

  Previous work to reach initial approach.

- chapter 3 Design and development.

  Modifications made during the carring out of the work.

- chapter 4 Evaluation.

  Asses the viability of the previous changes.

- chapter 5 Conclusions and future work.

  Outcomes drawn and how to improve.

Chapter 2

State of the art

2.1 Chapter overview

In this chapter we are going to introduce and briefly explain the main works on different topics that have led us in the current state-of-the-art. It is out of the scope of this chapter to delve into specific algorithms but rather our aim is to describe general multi-purpose methods.

2.2 Egocentric Video

Egocentric video has special features regarding other types of videos. In egocentric videos the camera is attached to the head of the wearer such that the resulting video is captured in first person point of view. These videos emphasize the relationships between the human and the environment, so here scene understanding also builds on how a person relates to different scene objects [1]. This thesis focuses on recognizing these human-object interactions (actions) in an egocentric video, in order to achieve region identification with these actions and visual features. Recognizing a region just by its visual features is very complex, since they can change a lot. In order to accomplish the task, it could be interesting to take advantage of the fact that the same set of actions are carried out in a region.

In this vein, it is worth highlighting the *epic-kitchens* work which provided the more relevant dataset for egocentric indoor video [2]. This dataset is currently composed of 100 hours of recording in FullHD and over 20M frames. The dataset encompasses egocentric videos of people performing natural tasks in their kitchens. There are 45 different kitchens recorded and 90K action segments.

The first paramount and promising system on the recognition of actions in the epic-kitchens dataset tried to model the hand grip and the object features with computer vision techniques. With this information it was proposed to infer the action with a single image and one model. First of all, the authors explored the relation between a grip and the object features, to asses if the acquisition of one of them could be used to accelerate the acquisition of the other. Then, authors proposed to detect the actions based on that each kind of data (hand grip and object features) provides complementary information of a specific action. The model is composed of three stages: first a visual recognition layer is used to obtain grip and object features, second a Bayesian net is used to explore the relation between grips and object features and finally a fully connected layer is trained to detect the action [3].

After the previous approach, many other well-functioning approaches were developed for action recognition in egocentric video. To better explain them, they will be organized in a classification based on the architecture they use. It is important to note that both, the previous approach and those that will be explained below, carry out action identification in egocentric videos but not focused on region identification. *Ego-topo* [1] (our baseline system) uses object and scene's appearance with the action detection explained to identify the regions as we will explain later in Section 2.4.

- **Two flow nets**: A relevant work in this line [4] used two CNN streams, one of them analyzes motion and the other appearance. The second one gets the appearance by segmenting hands and looking for the objects in the frame. With this information the system is trained to recognize actions and the human-object interactions. This system uses the video as a set of frames, not considering the temporal correlation.

  Another work and one of the most relevant papers on this architecture obtained human action and where he looks(the gaze) with an egocentric video. It models the gaze with the distribution of a probability given by deep learning units. The gaze, in egocentric videos, gives information about the action to be carried out, for example if a user is looking at his hands, he is probably holding something. The union of the gaze information and the visual features is used to obtain the action. For this purpose, the network has two flows, the first analyzes RGB frames and the second the optical flow [5].

- **3D convolutional models**: here the action recognition is performed by an end-to-end net, in which the full video is inserted. These networks can even predict future actions. The problem of this type of nets is that they are very difficult to train for proper results [6].

- **Recurrent Neuronal Networks (RNNs)**: Most systems are LSTM methods (a special type of RNN) and they take advantage of the special properties that the RNNs have. One remarkable system employed two LSTM nets, one of them to sum up the past and the other to compute future actions. The whole video is pre-processed to take RGB video information, optical flow and object features [7]. There are other systems [8] very similar to the previous one.

2.3 Representing the scene in a structured way

The aim now is to create a structured representation, as a graph, to encode more information about the scene present in the video. This allows to have a rich representation of the scene, where not only the objects and the actions can be observed but also other information such as the order and relationships between them. The section explains the transition to get these nodes to represent regions of the scene.

Here a state-of-the-art work in the creation of structured video representations is explained. The work considers that for a person to recognize he is taking a book is important to model the spatial-temporal dynamics of moving and the human-object interactions (the use of the object for the human) so it defines the nodes and their relations with both of the above information. Therefore, objects can be linked according to their spatio-temporal proximity or due to their similar semantic features (e.g.,

regarding human use), unless they are far in time or space. The architecture is as follows; First, the video is fed into a 3D convolutional model to get the features and then, a Region Proposal Network (RPN) is used to obtain from the features the important regions of the image. Finally, the system uses a *ROI Align* (operation to acquire the features of these new important regions). The features obtained by the ROI align are used to create the graph. For the creation of the graph a Graph Convolutional Network is included [9].

Alternatives to improve this baseline method are:

1. Including movement primitives throughout the video for a better action detection. The movement primitives are nothing more than local convolutional features. In this work they also added a spatial-temporal pooling to get better results [10].

2. Relying on memory based methods that keep a recurrent state in the net to bring previous information to the current frame. A pioneering work in doing this, brings past information thanks to a 3D convolutional model and a RNN. The RNN brings information on actions already analyzed by the 3d Net [6].

3. Using 3D convolutional model with a bank of features to give temporal context, as humans also need context to be able to recognize. A work in this line [11] proposes to use a long-term feature bank to model the context.

All these methods just account for scene objects and have created relations between them based on their spatial-temporal proximity and the features of the human-object interactions. Later, some methods added context and past information to improve results. Differently, baseline system for this thesis *Ego-topo* is not object based, it has a human-centric approximation,i.e., works according to how a human uses the space. So, the nodes created by *Ego-Topo* are regions of the space and their connectivity depends on how a person moves across them.

2.4 From objects to regions

As we said, previous relevant work on these structured methods is very diverse but mainly focused on objects but *Ego-topo* works with regions. Here, we sketch the evolution between objects-wise and region-wise methods.

At first there were works [12] [13] that learned to model the manipulation of objects. In the first one the authors considered that objects are manipulated and their state changes after a human action. The method identifies the state of the object and the action associated with the state change. In the work temporal consistency is assumed, that means an object passes from one state to another with an action in between. Groups of objects can be associated as a same region depending on the actions performed in between. For example, if there is a state representing an empty glass and another that represents a full glass, with the action of "filling" as a union, we could consider that this union of objects and actions has occurred in the same region. The second paper is similar but it paid more attention to how the subject grabbed different objects to detect the objects.

Secondly, another interesting idea is to explore how the pose of a user can help to identify an object. Each object has different functions and different individuals can use it in different ways. The original work [14] tried to make a recognition of the object

functions and group objects with similar functions, this group could be considered a kind of region. It described objects by the human pose when grabbing them and their visual features. A realistic human pose detector and YouTube videos were used to train the system.

A pioneer work that began to consider regions instead of objects [15] proposes to estimate the environment actions making use of the similarity between objects and scenes. Action maps are estimated by means of the known functionality of different regions. These action maps codify the option of carrying out different actions on several regions. It envisioned that actions can be associated to different regions, this is an important concept used in *Ego-Topo*. The difference with the *Ego-topo* approximation is that *Ego-Topo* does not only use the object and scene's appearance to get the action maps with which the regions are identified, *Ego-topo* uses object and scene's appearance with the action detection explained in the first section to identify the regions. Therefore, it makes use of the visual characteristics and the key point of the egocentric videos that is the action detection.

2.5 Structured region video representations: Baseline system and previous work

As just explained, *Ego-topo* performs object detection with the use of actions and visual features, achieving the best results in this field. It also makes use of structured representations to organize these regions and the connections between them. Nodes are the regions detected and the edges represent how they interconnect according to the video. Thereby the final graph is built, the use that will be given to the graph in this thesis is to know where a user has been and for how long.

Besides *Ego-topo*, there is one previous work [16] on this topic with egocentric video but the acquisition of the different regions in the video is not totally automatic. First, a person defines the different regions of a area and then when he/she comes again to the area, the system automatically detects them, but the process needs the first identification. When it has this first identification of all regions if an area appears in the video it only compares with the features of the different initial regions to detect the current one.

Both systems can determine the regions through which the user goes and consequently how long he has been there but *Ego-topo* does not need this first manual identification, which turns it a more scalable method. *Ego-Topo* is explained in detail in section 3.2

Chapter 3

Design and development

3.1 Chapter overview

In this chapter, first we explain the *Ego-topo* system. Then, we describe the process performed to improve its results and enhance its functionality.

3.2 Baseline method

The starting point is *Ego-topo*. It is a work published in 2020 by Facebook researchers [1]. As already explained in the previous chapter, *Ego-topo* represents the state-of–the-art for automatic place registration. We envision that, by combining the place registration of two users, it is possible to know if they have passed by the same regions, which is one of the goals of this project.

First of all, it is necessary to explain the baseline method to understand how it works and the subsequent improvements. *Ego-topo* tries to arrange the video into a graph with *activity regions* and the visits to these regions throughout the video.

3.2.1 Aim of the method

The method looks for the most relevant regions of the environment for human action and then associates to these relevant *action regions* the interactions carried out there. To achieve this, they need long egocentric videos where the conditions are not constantly changing. For example, the method would not work in a video of a city walk, where the places that appear in the video are different throughout the whole video. If the video is too short that it does not give different regions time to reappear, it will not work either. Therefore, ideally, we need long videos in which the different places reappear and can be analyzed for a sufficient amount of time.

3.2.2 Stages of the method.

The original work has three main stages:

1. Place Registration.
2. Inferring Environment Affordances.
3. Anticipating future actions.

For the objective of this Master Thesis we are going to use only the *Place Registration* stage as the baseline method, but we also briefly explain the other two parts.

Place Registration.

The purpose of this stage is to obtain different activity regions. To attain the objective an initial option would be to acquire the regions with visual clustering or geometric partitions. But as the *Ego-topo* authors explain, visual clustering is not enough because the features are biased by the moving objects in the scene, and the region detection should not be conditioned by these circumstantial features. Geometric cues can be hardly used because on egocentric videos there are very fast camera movements that do not let the method work correctly. Furthermore, these types of approaches do not take into account what regions are relevant for the human.

Construction of similar and dissimilar images. To handle these challenges, a localization network was proposed. This network was trained with *similar* and *not similar* pairs of frames:

- Frames are similar if:
  1. They are under the same action label or they are temporarily close.
  2. The homography that relates the two frames agrees with at least 10 key points correspondences. A homography is a geometric transformation that relates corresponding key points between two frames (images of the same scene captured from different points of view). In the Figure 3.1 there is an example of key point correspondences. If the frames have more than 10 key points in common that are inliers of the estimated homography, they are considered images of the same place.
- Frames not similar are the ones that do not meet either of the two requirements.

Training of the localization network. To train the localization network, authors propose to introduce pairs of similar and not similar frames. The network is a *Siamese network* (to be able to introduce pairs of frames) with a ResNet-18 skeleton. The siamese network is followed by a five-layers perceptron. The entire architecture is mainly inspired in [17], but the difference here is that the learning is carried out by the actions (since similar frames are frames under the same action label) and not only by visual features.

Graph creation and frame association process. Once the network is trained, a graph is created where each node is a different region and the edges joining the node define the relations between them according to the video sequence. For example, if a user first goes to the paper bin and then to the fridge, these two nodes will be joined by an edge.

The localization network is the fundamental part of the method, as the similarity score resulting for its evaluation defines how frames are associated to the different nodes and grouped in visits. The association process is as follows and it is graphically represented in Figure 3.2:

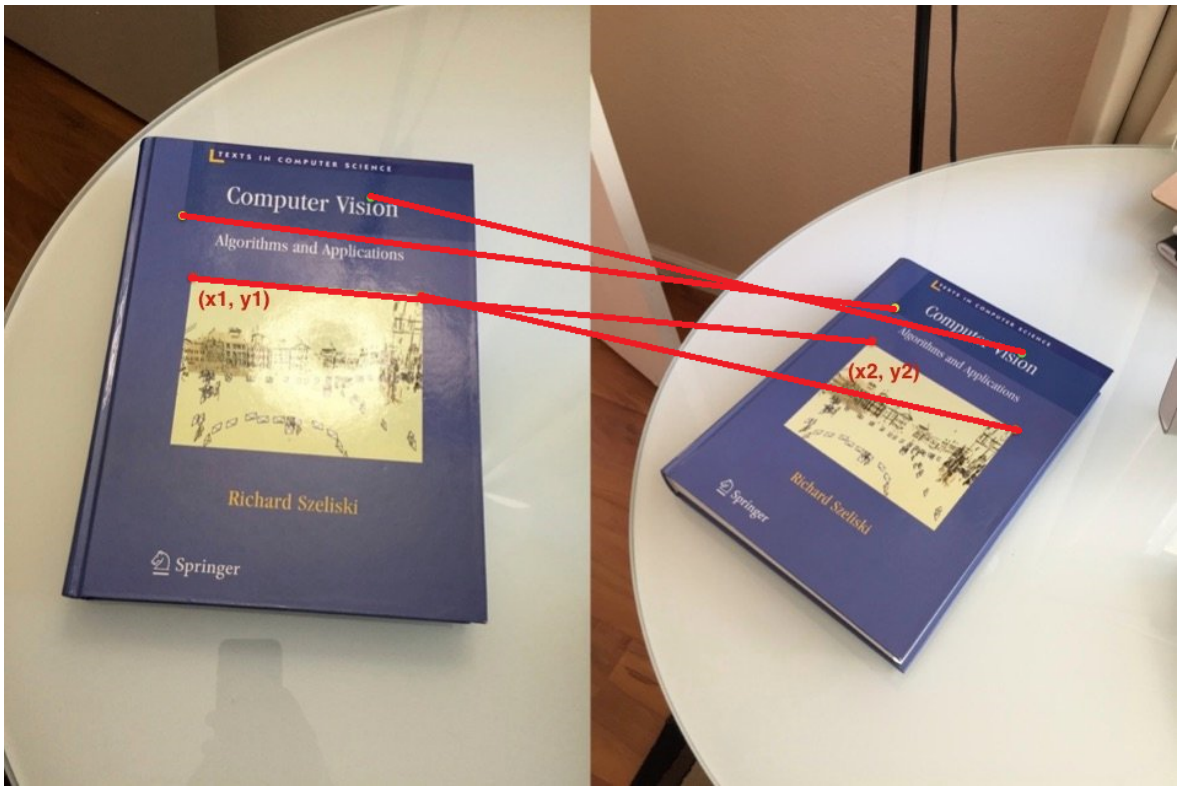

Figure 3.1: The homography between the two views of the book can be obtained from the four represented correspondences.

- The N center frames of each visit in each candidate node are selected, for example if the visit goes from frame 50 to frame 90 it selects N frames around frame 70 (the central one). If a visit has less than N frames, all of them are selected. So, if the node has M visits, the maximum total number of frames taken is $M \times N$, let's call this group of frames *NodeFrames*. Each node has its own *NodeFrames*.

- Then, the algorithm compares a window around the current frame with every visit frame selected on each node(the *NodeFrames*). The comparison is carried out frame by frame (all against all) yielding an overall similarity score by averaging all the comparisons. As we will explain in Section 3.6 not all visits are used, but this will be skipped for now.

- The network then returns an overall similarity score between 0 (totally dissimilar) and 1 (totally identical) for every node. The highest similarity score indicates the most similar node (region) to the one represented in the current frame.

- However, the association of the current frame to the most similar node is not straightforward. There is an upper threshold and a lower threshold against which this similarity score is compared:

    1. If the score is under the lower threshold for more than five consecutive frames a new node is created using all the consecutive frames under the threshold as an initial visit.
    2. If the similarity score is above the upper threshold and the second highest similarity score (against another node) is not closer to the highest similarity

score more than 0.1, the system creates a visit to the corresponding node with the consecutive frames that exceed this threshold.

3. If the value is between the thresholds or if the two highest scores are above the upper threshold but the difference between their similarity score is less than 0.1, the approach does not update the graph (skip zone).

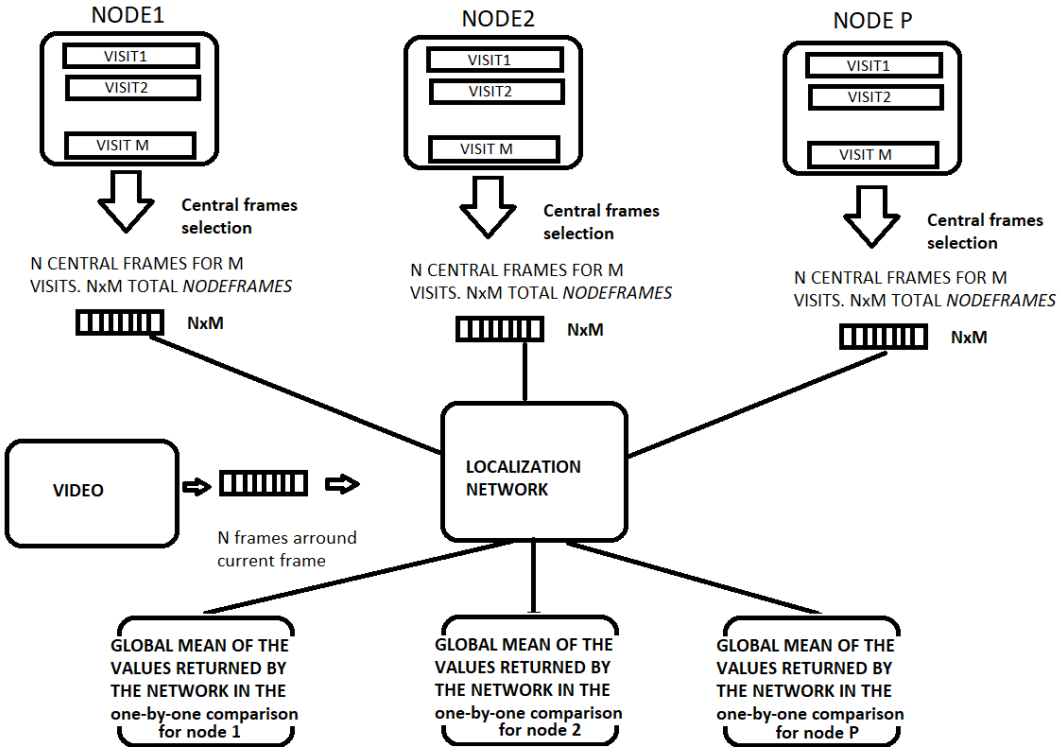

Figure 3.2: *Ego-topo* algorithm operation

Inferring Environment Affordances.

This stage deals with the association of regions of different videos that have the same functionality. For example, linking the sink in different kitchens despite not being the *same* sink. To accomplish this, each node uses a classification network of actions and objects which gives a distribution of (action,object). Using these distributions they aim to associate nodes of different kitchens. It is certainly an interesting part but out of the scope of this master thesis.

Anticipating future actions.

This stage deals with the strategies followed to predict the actions of the last 75% of the video by just observing the initial 25%. Again is a fascinating part but also out of the scope of this project.

3.3 Place graph updating

The objective of this thesis is to be able to recognise if two users have passed by the same place or not to trace possible indirect contacts in the scenario of a contagious disease. To carry this out, the original *Ego-Topo* Place Registration strategy (see Section 3.2) is not sufficient. At least, we need to define a new strategy for being able to start from a previous graph and add visits on the existing nodes if a new user moves through one of the previously registered nodes (regions). In this manner, there is enough information to detect a possible contact between two different individuals.

To attain this, we start from a previously created graph with the information of the nodes and the frames of the video itself, and use its nodes and visits to feed the localization network when needed. This has only been implemented with two videos for storage and computational reasons.

Finally, the system was upgraded to be able to create these combined graphs composed by two videos. Several changes among the code were carried out and the incorporation of the new video name as a parameter if we want to update it with a previous one.

The main aspects changed in the code are:

1. The video graph structure is saved and load with pytorch *torch.load* and *torch.save* functions.

2. If the user is asking for a combined graph, the code loads the graph to be updated and the video it comes from together with the current video to be analyzed.

3. If the system is in *combined-graph* mode, it starts using the previous graph and every previous node when the frames of the new video are computed.

4. In the updated graph, to avoid frame confusion problems, the length of the previous video is used an offset, i.e., it is added to the number of the frames of the original video so that there are no frames with the same number.

3.4 Enabling the use of external videos

This is a very small change compared with the previous one, the system is prepared to use only videos of the Epic-Kitchens or EGTEA+ datasets because it needs the length of the video and the frame rate as meta-information. The system was adapted to also be able to operate on external videos introducing its length in number of frames and the frame rate as a parameter. The reason to do this is to understand how the system works in external videos which have different conditions. Furthermore, it also permits the hypothetical setting up of the method on specific conditions defined *a priori*. For example, a developer working with the code wants to check if the system is working properly in the event of a lighting change in the scene. This way he can record a video, input it and analyze the behaviour. Combined with the previous upgrade, the method can also update graphs from external videos.

3.5 Limitations and solutions

Once these upgrades were developed and assessed, we evaluated the behavior of the method using different sets of challenging example videos in order to determine the system performance and drawbacks and to ensure that it is able to operate correctly in the envisioned application. The next sections will contain the two errors found in these tests.

3.5.1 First limitation

We start by creating a location graph corresponding to a video from epic-kitchens dataset (P02_01, see Figure 3.3 left for an example frame), and evaluate the updating of the created graph using information from a different video of the same kitchen (P02_02, see Figure 3.3 right for an example frame). They are really different videos: the first video is about preparing a coffee and the second one is mostly washing the dishes, but both videos captured frames of the sink region. Therefore, we can expect that, in the combined graph, sink frames from the second video will be associated as visits to the corresponding sink node created by analyzing the first video.

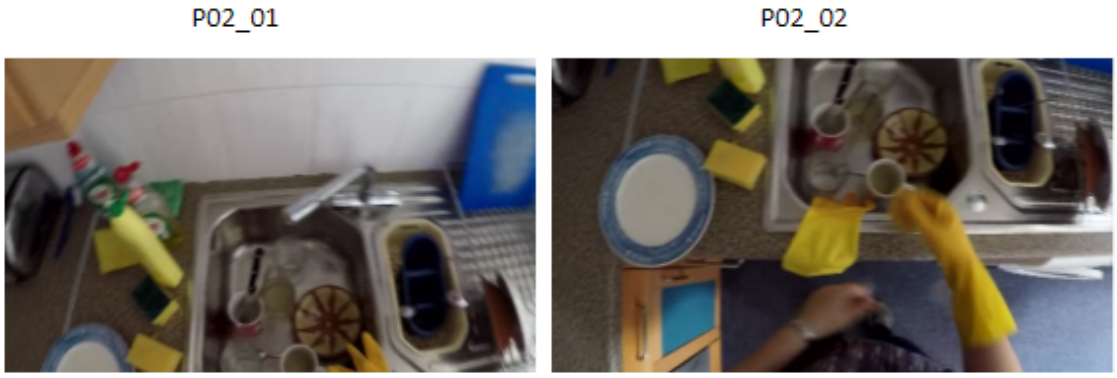

Figure 3.3: Sink on both videos

The initial graph created from video P02_01 is on Figure 3.4. The node to which the sink belongs is the second one, labeled as node “21”. This name is assigned according to the number of the first frame that was assigned to this node.

Now the graph is updated using the second video. To see what happens for the frames in the second video capturing sink region, we propose to analyze the scores returned by the network when a window containing sink frames is compared against each one of the nodes in the graph. This way, we can realize for each frame whether it is being assigned to the sink node or not. For example, in the Figure 3.5 the system is in the frame 1626 and the mean value for each node on each frame of the window around the frame 1626 is plotted. We are using the default window size proposed by the method, so it encompasses frames from the 1622 to the 1630. The score values obtained for these frames are averaged to yield an overall window-to-node comparison as explained in Section 3.2 (the final value would be the mean of the 9 scores). The Figure 3.5 shows how the score values for node 21 start to increase as soon as the sink appears in the second video.

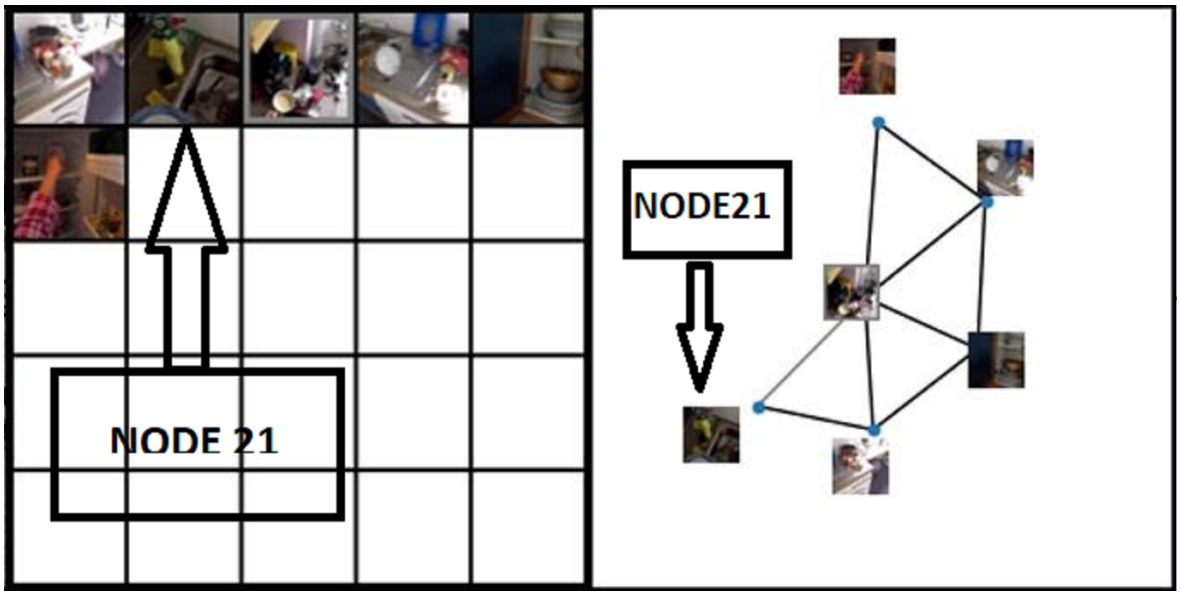

Figure 3.4: Initial graph for P02_01

This is hopeful, as it seems like the method is going to find the correct node, but we found out that the score got stuck and in a big part of the remaining video the value is between the upper and lower thresholds (skip zone). The Figure 3.6 is the typical graphic for the following 50% of the video.

This situation suddenly changes when the video focuses on another part of the sink, then the scores go under the lower threshold (for more than five frames) and the program creates a new node. The Figure 3.7 shows one frame of the region that caused the change and Figure 3.8 the similarity score evolution. From this point of the video we find that both nodes compete to get the higher value for the sink, but after some frames the new node *wins* and most of the sink frames are assigned to that node. Figure 3.8 shows the creation of the new node named 2121. The final graph with the new sink created node is in Figure 3.9

Why does the above event happen?

The node 21 is only composed by one visit from the first video, with frames of the visit being very similar to each other. In the second video the yellow gloves are an important part of the video content but, since they were not in the previous video, the score value obtained when compared these frames with node 21 is lower than the upper threshold. Additionally, in the second video the camera seems to be more inclined, so when the sink appears the wall behind it (that neither was clearly visible in the first video) is also captured. The subtle changes in appearance do not let the system add frames as new visits to node 21, inducing the stuck situation.

Then, another sink region slightly more different than the previous one appears and a new node is created, at the start these two nodes are very similar so sink frames have similar score values for the two nodes (21 and the newly created node). But when the new region that produces the node creation appears again, a visit is added to the newly created node. New visits also have parts of the sink, therefore when some visits are added to the newly created node it turns the principal node for representing the sink region. From this moment almost every sink frame is associated with this node.

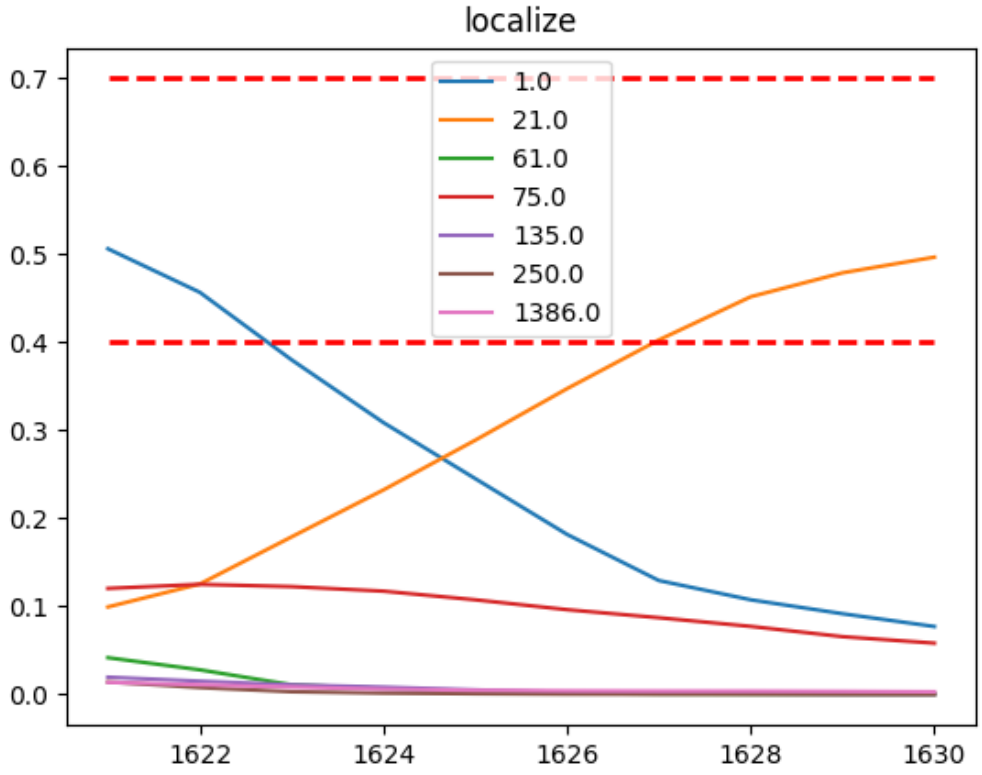

Figure 3.5: Initial sink graph score values

3.5.2 First limitation possible solutions

The possible solutions to this problem are varied, here few of them are sketched.

1. The problem is that, although the regions are very similar, little changes do not let the system assign frames to the corresponding node, because of a too strict metric. If the initial frames would have been associated with the correct node, then the system will have been able to use them to compare with the rest of the frames and the similarity score will have been increased. Thus, decreasing the upper threshold on the initial part of the video may fix this. However, this is a solution that only works if the "unidentified" region is at the beginning of the video, otherwise it would be necessary to specify where to decrease the threshold for each video. As it is not a general solution as we assume that the decrease of this threshold may be dependent on each video, we discarded it.

2. When some changes appear and the visit/visits in a node do not include representations of these changes, it is very rare that the similarity score surpasses the upper threshold in order to assign new visits, because the system computes the mean value of similarity to all frames from visits (what we called *NodeFrames*). This is not very effective; this way current frames (from window under analysis) need to be very similar to all frames in *NodeFrames*, quite a difficult duty. A possible solution is to change the mean by another metric more flexible like the median, this way the overall score for current frames will not consider all frames, but only the most representative or more similar ones, so the association will

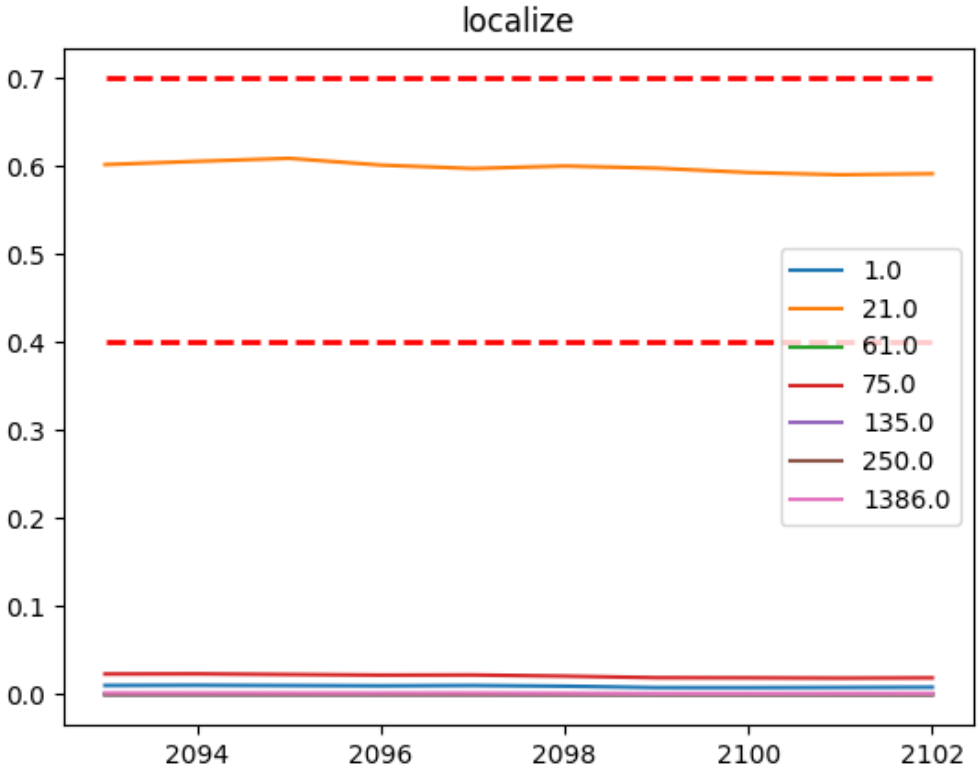

Figure 3.6: Rest of the sink graph score values

Figure 3.7: New sink region

be made easier. This one will be the solution implemented and is explained in Section 3.6.1. This option does not have the problem of being too video-specific.

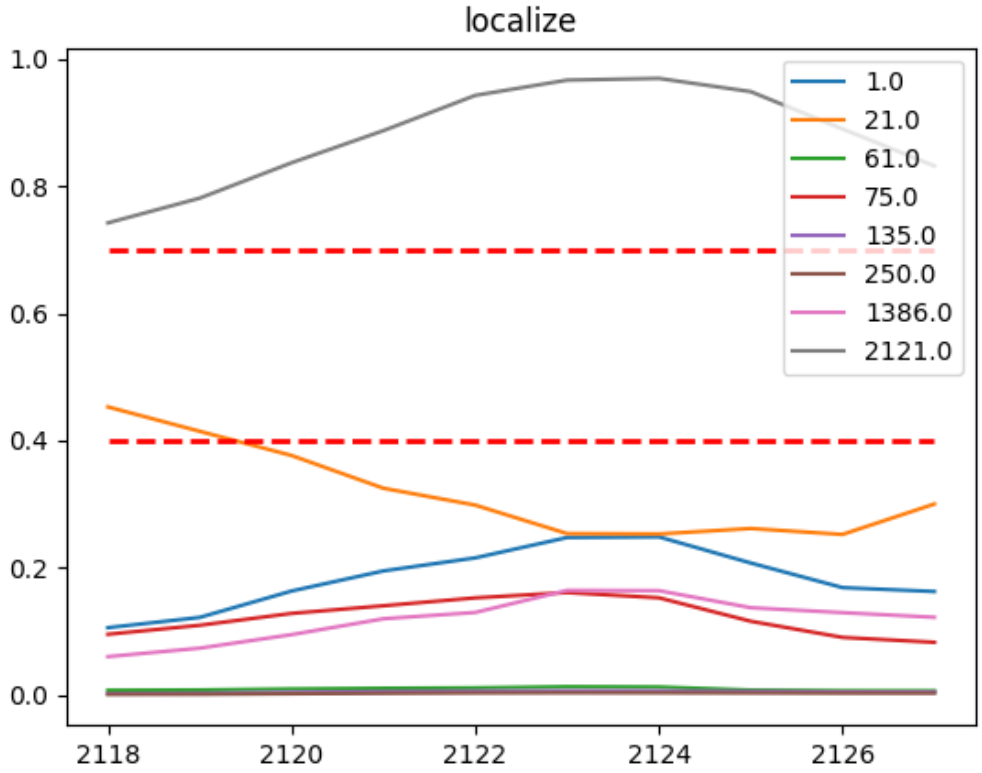

Figure 3.8: New sink node

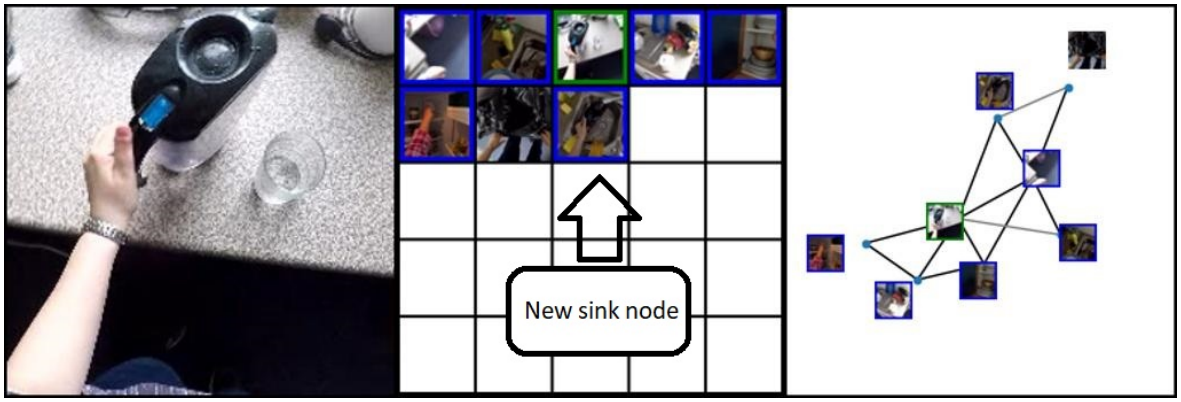

Figure 3.9: Combined graph for P02_01

3. Another solution would be to only take the more representative frames of each node to compare. In other words, select the frames chosen for each visit in a more intelligent way, not simply taking the central ones. This is an interesting option to be explored as part of the future work 5.2.

3.5.3 Second limitation

Then, we evaluated the updating of the graph created by analyzing the video P02_01 with the information provided by a external video of a different kitchen. In the external video there are approximately 4 regions (the sink, dishwasher, fruit platter, and the

countertop) see Figure 3.10 for example frames of these 4 regions in the external video. If the simple graph (without the updating strategy) is created, the method assigns these 4 regions to 4 nodes (see Figure 3.11 for the graph). Differently, in the combined graph, 4 new nodes for each one of these previous regions should appear but, instead, there are only 2 new nodes (see Figure 3.12) in comparison with the initial graph (see Figure 3.4). If we look at the graphics of the dishwasher frames (Figure 3.13), we realize that the system is not creating the region because it detects some similarity with other nodes. These nodes are from P02_01 video and they have no relation with the dishwasher. In the Figure 3.13 we can note how for some of these nodes (1, 61 and 75), high similarity scores are obtained for the dishwasher frames. It is important also to highlight that the score values for the other nodes are relatively high in comparison with the low scores obtained for not similar regions in the study of the previous limitation (Figure 3.6).

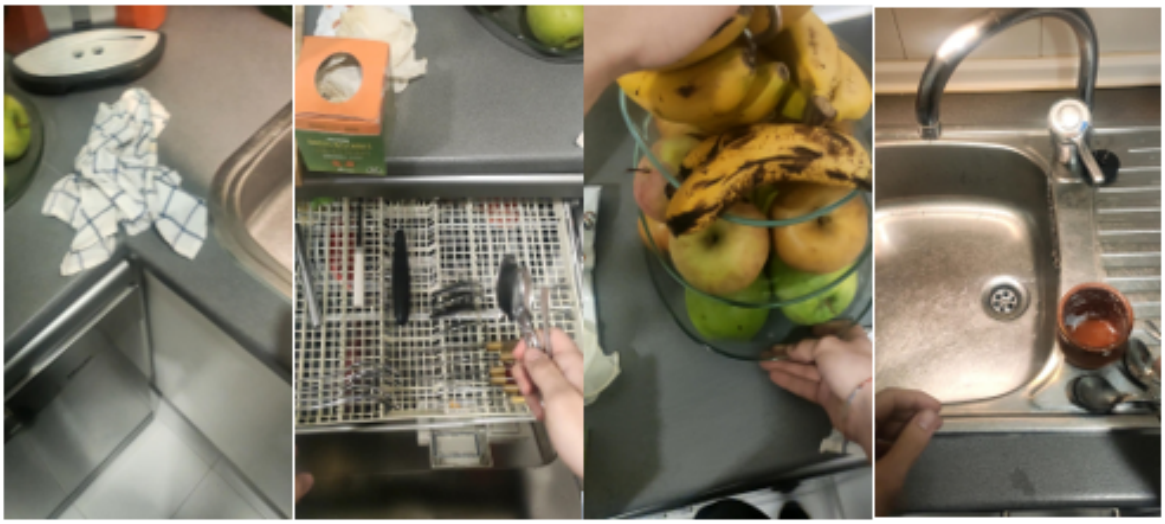

Figure 3.10: Example frames for the different regions in the external video

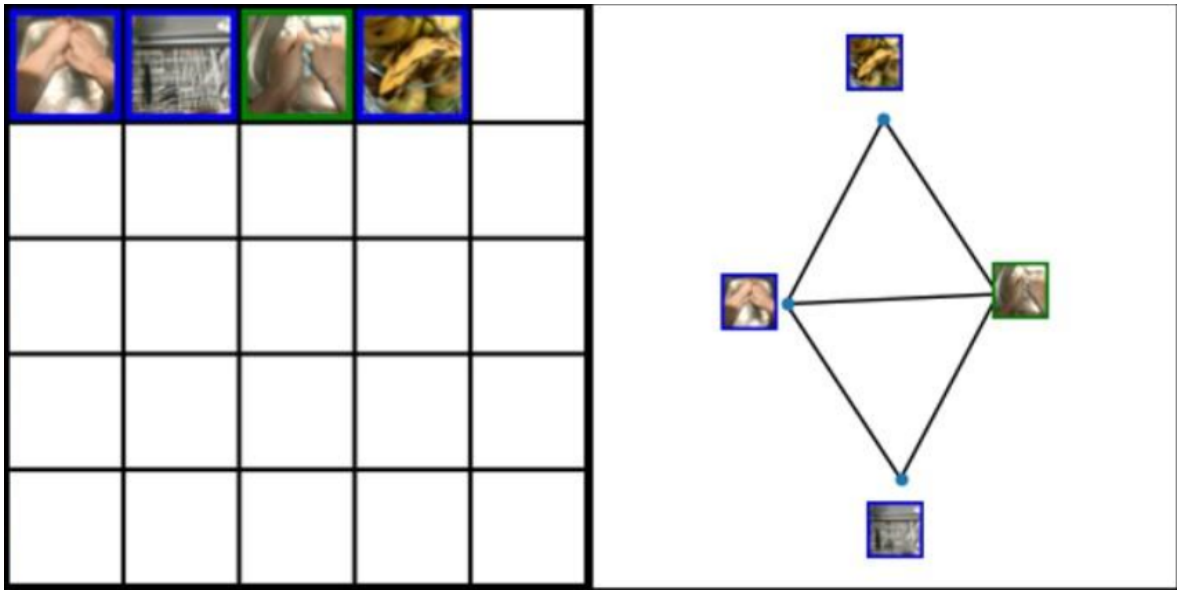

Figure 3.11: External video graph

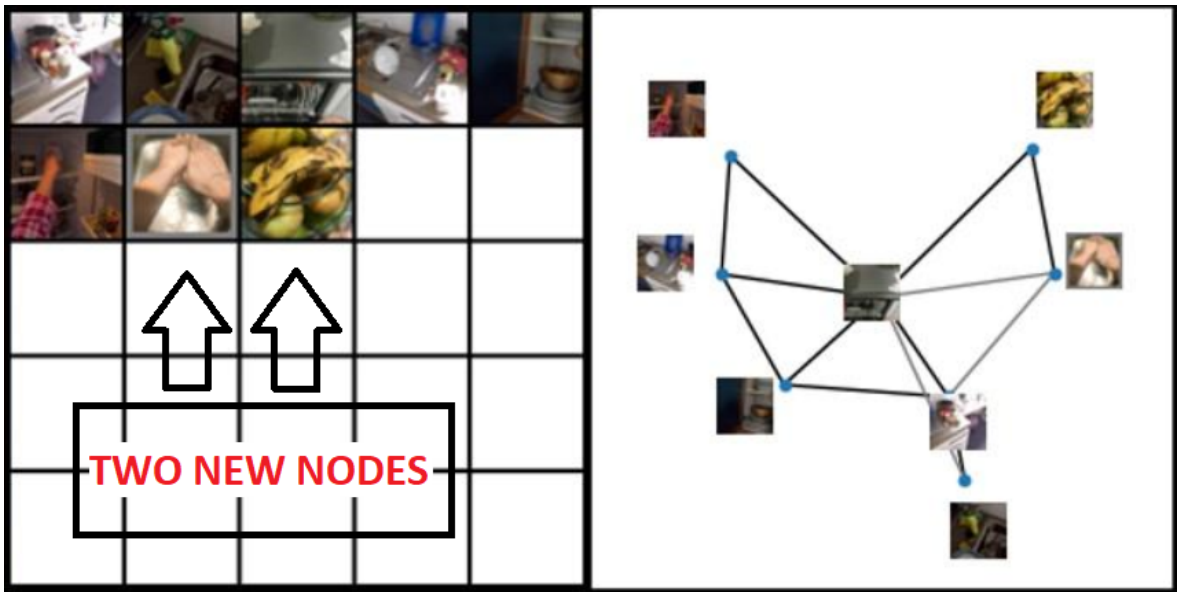

Figure 3.12: External video graph combined

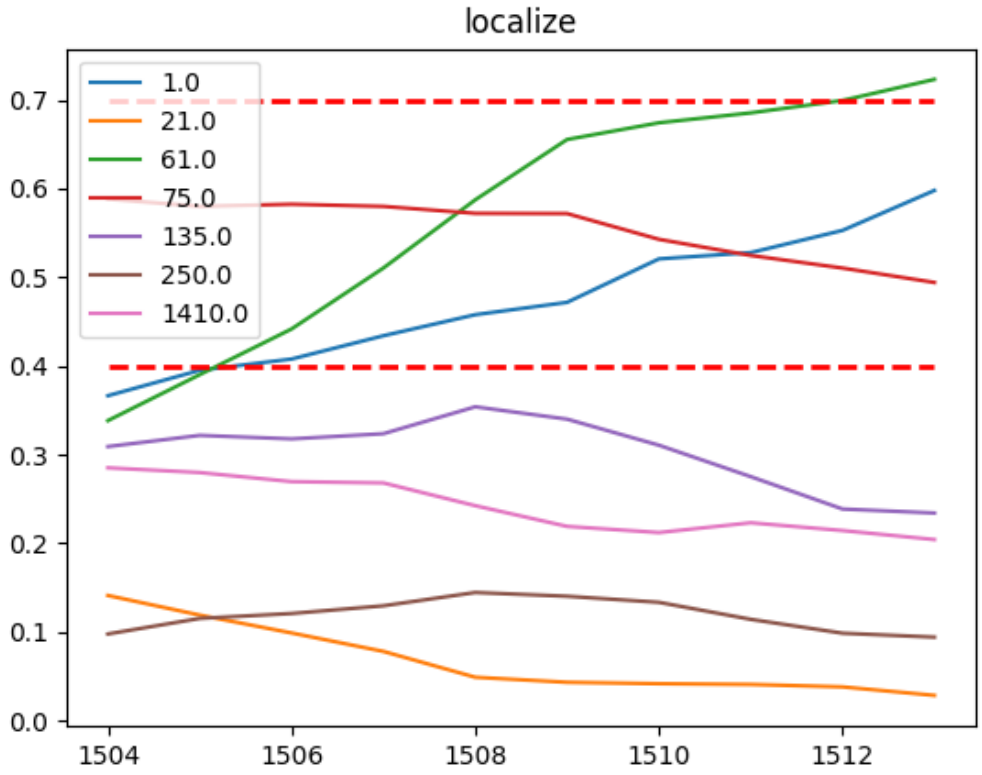

Figure 3.13: Dishwasher frames graph

Why does the above event happen?

This is an effect of the similarity network, because it is not trained to distinguish between the regions of the external video and the epic-kitchens video regions. Therefore, it confuses regions despite being totally different, the network only distinguishes really

different regions like the fruit platter, which has very different colors and shapes that any other node, and the sink because is very homogeneous and easily recognizable. But with the other regions the network yields high score values for too many nodes.

3.5.4 Second limitation possible solutions

The preferable solution is to retrain/fine-tune the similarity network so it can distinguish between regions of the new videos that it has never seen. Again a metric change could also help to partially handle this limitation.

3.6 Method solutions developed

The previous limitations affect the behavior of the application, causing errors in the creation and updating of the graph.

- The first limitation 3.5.1 is the main problem of the system, when the same area appears again in the video, it is frequently not detected correctly and the system remains in the skip area. After a while, this zone is associated with a new node.
- The second limitation 3.5.3 is more about the network. It does not let the system create combined graphs of videos captured under different conditions different to the ones used to train the *localization network*. This is a problem that will remain always there is a domain change, but is not a topic to be dealt with in this thesis. Despite this, if there is any way to smooth it out, it will be tried.

To find possible solutions the code has been thoroughly analyzed, particularly the method used to obtain the similarity score for each frame and each node. The basic operation to obtain the score values was explained on Section 3.2.2 so it will not be explained again, but a detail that was omitted in the previous description for the sake of clarity is included here.

As already explained, the system is designed to select the central frames of each visit. But, regarding the visits, the algorithm does not use all the node visits and neither there is an *educated* criteria for selecting visits, it takes always 20 uniformly sampled visits for each node:

1. If there are less than 20 visits, visits will be repeated uniformly until 20.
2. If there are more than 20 visits the system will take 20 visits uniformly sampling the vector.

Then, it takes the frames around the center frame of these visits, following the order given by the previous operation, no matter if they are repeated, and concatenates all of them in a vector called key_frames. These are the frames used as *NodeFrames*, to compute the overall score as explained.

Is the above behaviour necessary?

There are two cases:

- The number of visits is less than 20:

  - In case the number of visits is divisible by 20 the operation is totally useless; it just increase the operation overload of the method. For example, if the number of visits is 2, the vector key_frames will be composed by the 10 times replication of the center frames of visit 1 and the 10 times replication of the center frames of visit 2. Therefore, it will have 10 times repeated the similarity scores of each visit. After the mean calculation, the final similarities are the ten-times repetitions of the two visit similarities.
  - In case the number of visits is not divisible by 20, the operation has repercussions on results. Due to the sampling scheme, the middle visits will be repeated more times than the first and the last, i.e. if the number of visits is 3, the first would be repeated 5 times, the second 10 times and the third 5 times. It gives more relevance to middle visits but only in case the number of visits is not divisible by 20.

- The number of visits is higher than 20:

  - The system will choose visits without any criteria only by sampling uniformly the visits.

This operation is useful when the number of visits is greater than 20. When the video is too long the number of visits grows a lot and the execution time increases exponentially, hence, it is necessary to limit the number of visits considered. But probably the uniformly selection of these visits is a process that can be improved.

In the event that the number of visits is less than 20, the difference is when this number of visits is not divisible by 20, here the method gives more relevance to central visits. Giving more importance to central visits and only in some cases does not seem very logical. Why should the second visit have more weight than the first one?

Finally if the number of visits is less than 20 and it is divisible by 20, the selection process is totally useless.

In short videos the operation will increase the computational cost, because the number of visits is usually less than 20, and with this behaviour the system repeats them until the 20 visits. Moreover, in some cases it will add some noise due to the fact that it gives more relevance to center visits. Therefore, we decided to get rid of this selection process. In the case of the videos that we are analyzing, since they are short videos, they will not suffer the computational problem of exceeding 20 visits. Anyway the computational problem for long videos remains, in Section 3.7 we explain how the computational problem is tackled.

The elimination of this process made slight changes but did not solve any of the described limitations, so we explored alternative solutions.

3.6.1 Metric change

As we have explained at the beginning of this section, the first limitation is the main problem to be faced. It is too frequent and it harms the performance of the system, especially for one of the objectives of this thesis: *Detecting possible contacts of different users*. With respect to the second limitation, it is positive if its effect can be reduced, but it is not a priority for this thesis.

The main problem to solve the first limitation seems to be the metric as we have already explained in the Section 3.5.2. By using the *mean*, to associate a new window to

a node, the method requires the frames in the window being similar to all the frames of all the node visits. This also affects the node heterogeneity, as this association process tends to create homogeneous nodes where adding a visit is quite a difficult task even if there are just subtle appearance changes that differentiate the frames in the window from those in the visits.

In the study of the first limitation (3.5.1) we stated the problem: the total mean scores are sometimes near the upper threshold but rarely over it so the system can not add visits to the node. To explore if other metrics can be used to solve this problem, alternative a more detailed analysis is needed.

Specifically, we need to account for each window-visit similarity score (and not only for the averaged ones), so we can analyze more in depth what is happening. These similarity scores for the situation analyzed in Section 3.5.1 can be observed in the Figure 3.14. In particular, the figure depicts the comparison between the center frames of the visit in the sink node and the window around the frame under analysis. These sum up to 9 graphics in the same figure (one per frame in the window around the current frame). The number of points (X axis) depends on the amount of frames in the visits (9 per visit or less if the visit has less than 9 frames). To clarify, the mean of each plot, results in a point in the per-frame scores depicted in the figures of Section 3.5.1.

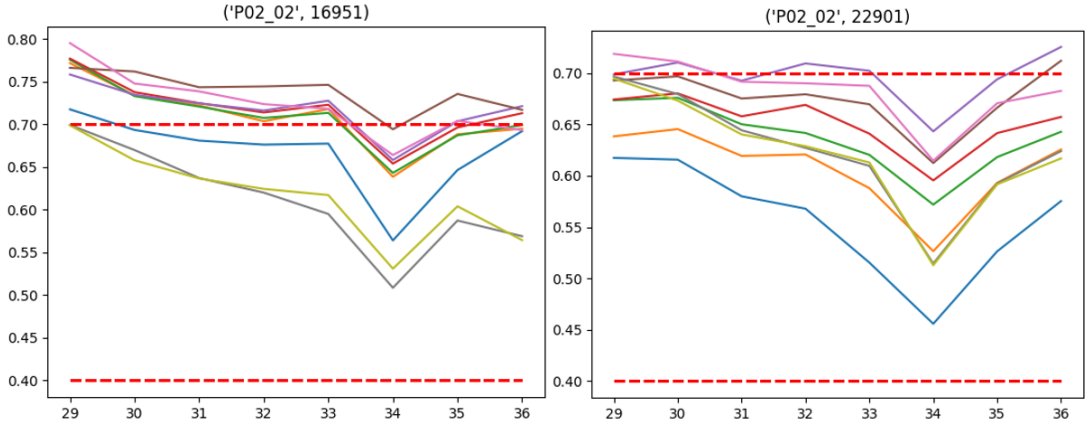

Figure 3.14: Specific score values for the first limitation

In the Figure 3.14, the left graph shows the score values for a frame in which the system almost associates the frame to the node. Differently, on the right side we include a graph with typical scores that result in the stuck situation previously described. When we analyze this figure, we realize that despite the mean value not being over the upper threshold, a big amount of individual scores exceed it. As it was thought at first, the metric is maybe too strict for associating regions. The solution to the problem can be sought by changing the metric and evaluating the benefits and problems of each alternative metric by analyzing the changes produced by it in this *test example*.

First option: using the maximum

The maximum method (max) consists of using only the highest value of all the graphs to associate i.e. the most similar comparison between the window of the frame being

analyzed and the central frames of the visits. In the *test example* 3.14, with the max method these two graphics would associate frames to node 21 because both of them have some scores over the upper threshold. New associations would help to increase the score values in the following comparisons, because the inclusion of this window as a new visit that will be used to compare with the following frames, including potential new features that the previous visit may not have. Therefore, this may be advantageous and may assist to avoid the creation of new nodes but instead aid the association of new visits to the existing one.

An advantage of not using the mean is that the negative effect of outliers is reduced. Outliers are frames taken as visit frames but with different features with respect to the other visit frames. They make the system returning altered mean values. We can see this effect in the *test example*; in the Figure 3.14 frame number 34, always gives a peak down. This is one of the potential problems of the comparison: frames that have different features because of a fast movement of the camera to another point, and are associated with the visit. The Max comparison partially solves this problem because it does not consider this value to associate, but alternative options could be also valid. For example, avoiding the acceptance of frames that are different from the visit, e.g., if in the window frames, the frame under analysis is very different but is accepted because the frames around are similar, this specific (outlier) frame can be removed from the visit before creating it.

We need to make sure that using the max will not be very noisy and will associate random frames to nodes that do not correspond. Using only the max point for the 9 graphics is a potentially risky idea because, as we have seen, there are peaks that may alter the results. One option would be to take two or more *max scores* from each plot (each frame scores), and then the mean between these points. This way the problem of the peaks is partially avoided as these are smoothed. But there is still a flexibility problem: if the max is used to get the final score, it will be easier to associate, but it would be almost impossible to create a new node since the metric takes the highest values, and to create a new node the value needs to be under the lower threshold. Therefore, this strategy may inhibit the creation of new nodes.

Second option: using the median

This metric affects upper and lower thresholds in the same way, without harming one of the decisions. This option uses the median of all the similarity scores calculated for the decision, so if the majority of the similarity scores are above the upper or under the lower threshold it is enough for driving the association process. This is also a good choice to avoid outliers.

To test this idea, let's see again what would happen in our *test example*. Here (Figure 3.14), only the first case will be added as a new visit. This case is the most favorable and is only found in a couple of frames, the rest have a behavior similar to the graph on the right and would not exceed the upper threshold with this measure. This could be enough because from this moment (the association of the frames corresponding to the graph on the left) the following frames could give higher scores thanks to this new visit. But it has a problem because until the frame of the left graphic where the association is performed, there are 1000 previous frames that result in lower scores than that required and won't be associated. On the other hand, in other videos we cannot ensure the generality of this example, i.e., we may have all the similarity scores

like the right one so no frames would be associated. That would not be enough and the problem would remain the same, therefore this metric can not be assumed.

Third option: using a combination of the maximum and the median

This option helps to associate by being more permissive than the median, but less than the maximum and without inhibiting the creation of nodes. For each frame in the window around the frame under analysis (each plot) if one frame yields a similarity score greater than the upper threshold, instead of doing the median, only the highest value is taken; if not, the median value is used. Finally, a global median between the calculated value for each frame in the window around the current frame (each plot) is performed. This is very similar to the above approach but it is a bit easier to associate new visits (which could fix the problem we kept having), it is a combination between the max approach and the median approach.

In the second limitation we have also observed that there are problems with creating new nodes, because the system does not have enough small scores to create them. The problem could be similar but in the lower threshold, so we perform an analogous change to the previous one; if some value is below the lower threshold only this value is taken.

Although, the second limitation could be smoothed in some way with this change, it is more a network problem: the net gives high similarity scores against a lot of nodes when performing the similarity between the new region and the previous ones. On the higher threshold, the network differentiates the region but does not give a similarity value high enough but here, regions are not differentiated so it is not a metric problem being too restrictive in the creation of nodes. But it is going to be implemented nonetheless, even if the system does not finally improve.

It is also possible that a comparison with the visits results in some similarity scores above the upper threshold and in some others below the lower threshold, in this case we will also consider the median value.

The results have been evaluated on the *test example*, but they were not as expected, in the sink region with these changes, instead of associating to the previous node, it creates another one, therefore the problem is not solved. We can appreciate this in the Figure 3.15. We realized that this change had made the association too permissive, the approach creates no sense nodes throughout the videos.

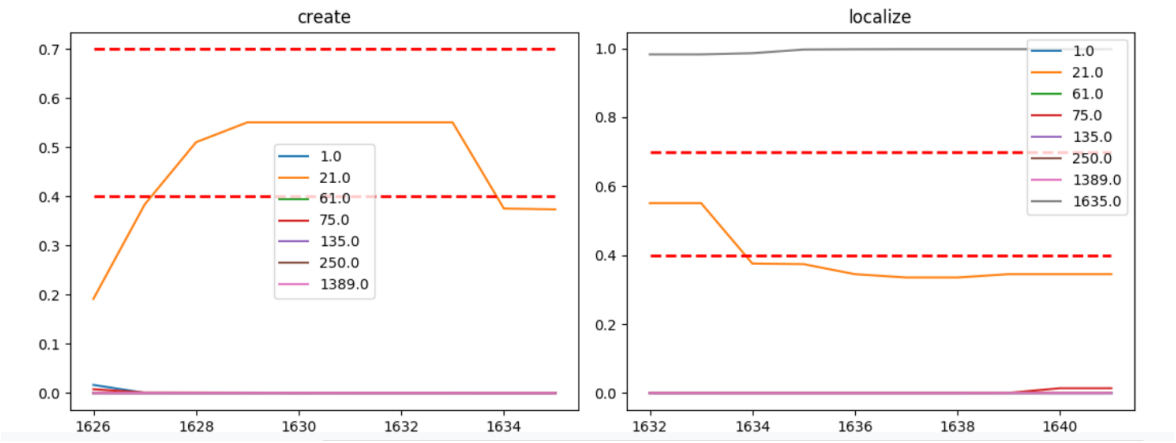

Figure 3.15: Scores graph for the *max and median* metric

Final option: using a combination of maximum and median with and intermediate value

In the previous comparison, the problem is mainly produced by the creation part that takes the smallest value if there is anyone less than the lower threshold. This change is counterproductive as the creation has become too permissive. For this reason, the last modification is removed.

As we described previously, the similarity scores sometimes have outliers (3.14). Outliers can cause a visit that does not represent real similarity to be associated erroneously due to the use of the maximum. To ensure that this association of a non-similar group of frames does not occur, the following modification was performed: To use the max value it is needed that the minimum value of the frame under analysis (the plot) is closer to the upper threshold than to the lower one. For example, in the *test example* (3.15), if the minimum value of the each plot is over 0.55 (closer to the upper than the lower), then the max can be used. Therefore, the max with this approach is only used if the frame (the plot) has a value over the upper threshold, and the lowest score of the frame is closer to the upper threshold than to the lower one. This ensures that when the maximum is used, every comparison has an acceptable similarity, and is not only an *outlier* causing false associations.

This is the final metric, the results for the *test example* as we can see in the Figure 3.16 are correct, after few frames the system finally associates the sink in the new video with a node of the previous one, and it does not create another one. In general, analyzing other videos, differences between the baseline and this last approach can be appreciated. Later in Chapter 4 we will quantitatively analyze if they are better or worse for a larger set of examples. The final graph is in Figure 3.17, and compared with the first one (Figure 3.9) it can be seen that the new sink node is not created. The sink frames are associated to the original sink node (*node 21*).

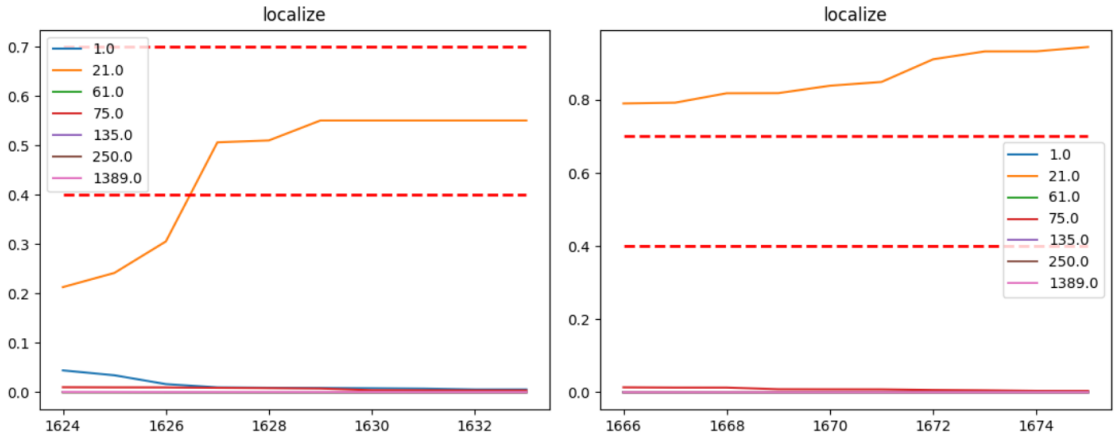

Figure 3.16: Scores graphic for the final metric change

A better metric for the first limitation has been apparently reached, but let's check this change on the second limitation to get an idea of how it has affected it. Results are similar but there are some relevant differences, as we can see in the Figure 3.18, the dishwasher region is associated with the fourth node in the previous video, which is not the same as before the comparison metric was changed, this is because of the new way to associate. Also, the countertop is now associated with the third and the

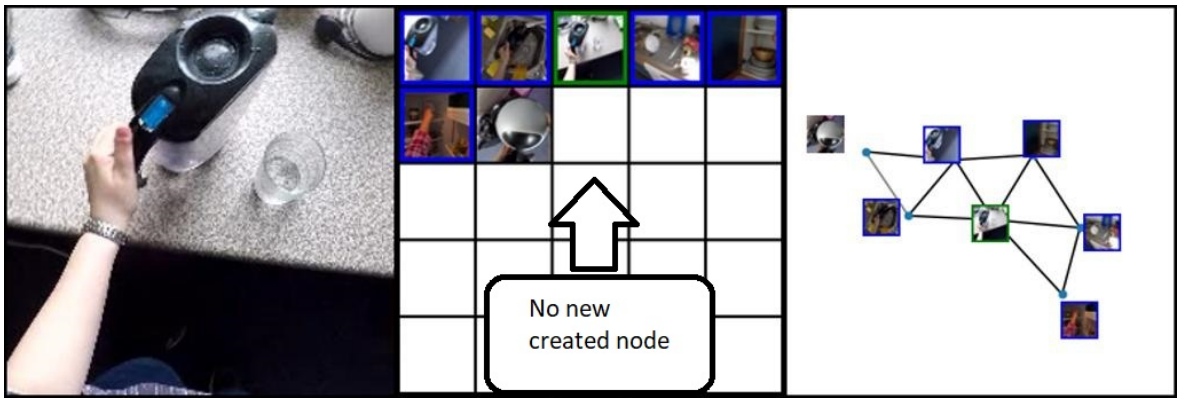

Figure 3.17: Resulting graph for the final approach

first node.

We can discern in the Figure 3.19 specifically the graphics to see why this is happening. In the left column we have the original and the new graphic for the dishwasher case. In the right column the same for the countertop case.

As we can see this is a net problem because it detects similarities with the nodes, and there is no relation between them, but the new association has slightly enhanced it. The ideal solution is a tuning (domain adaption or retraining) of the similarity network to be able to differentiate these new environments. With other improvements in the metric this problem could be smoothed out, but the problem does not come from the metric therefore there is no point in trying to fix it this way.

This result can be assumed, because the problem is not about the metric, but about the network. Furthermore this will not be the target study of the master thesis, in this master thesis, videos from the same place will always be introduced, so the error when introducing different videos is not critical.

3.7 Handling unconstrained searching time

In the previous section we discussed the elimination of the method part that always selects 20 visits, and we discussed the pros and cons. The main problem is that now the number of visits is not limited, and in the next chapter, a lot of videos will be used that are longer than the previous ones. With the original approach, in terms of time, for the longest video we used, the time went up to 15 hours, but with the new approach the same video took more than 35 hours. It depends on where the system is run, and the time available, but at least to meet our goal, and to achieve a relatively deterministic run-time that facilitates its use in other tasks, the number of visits will be limited.

To this aim, it is a must to limit the number of visits, so the time does not increase exponentially. The original random method is not the better option, it can be done in a more *educated* way.

To get more heterogeneous visits the following strategy was performed: a matrix is created in which each row and the column is a visit and the scores are the comparison with the corresponding visits, i.e. position (2,3) is the comparison between the second and the third visit. When a new visit is added, the matrix is updated. For each visit the mean comparison value against every other visit is computed, and from there the

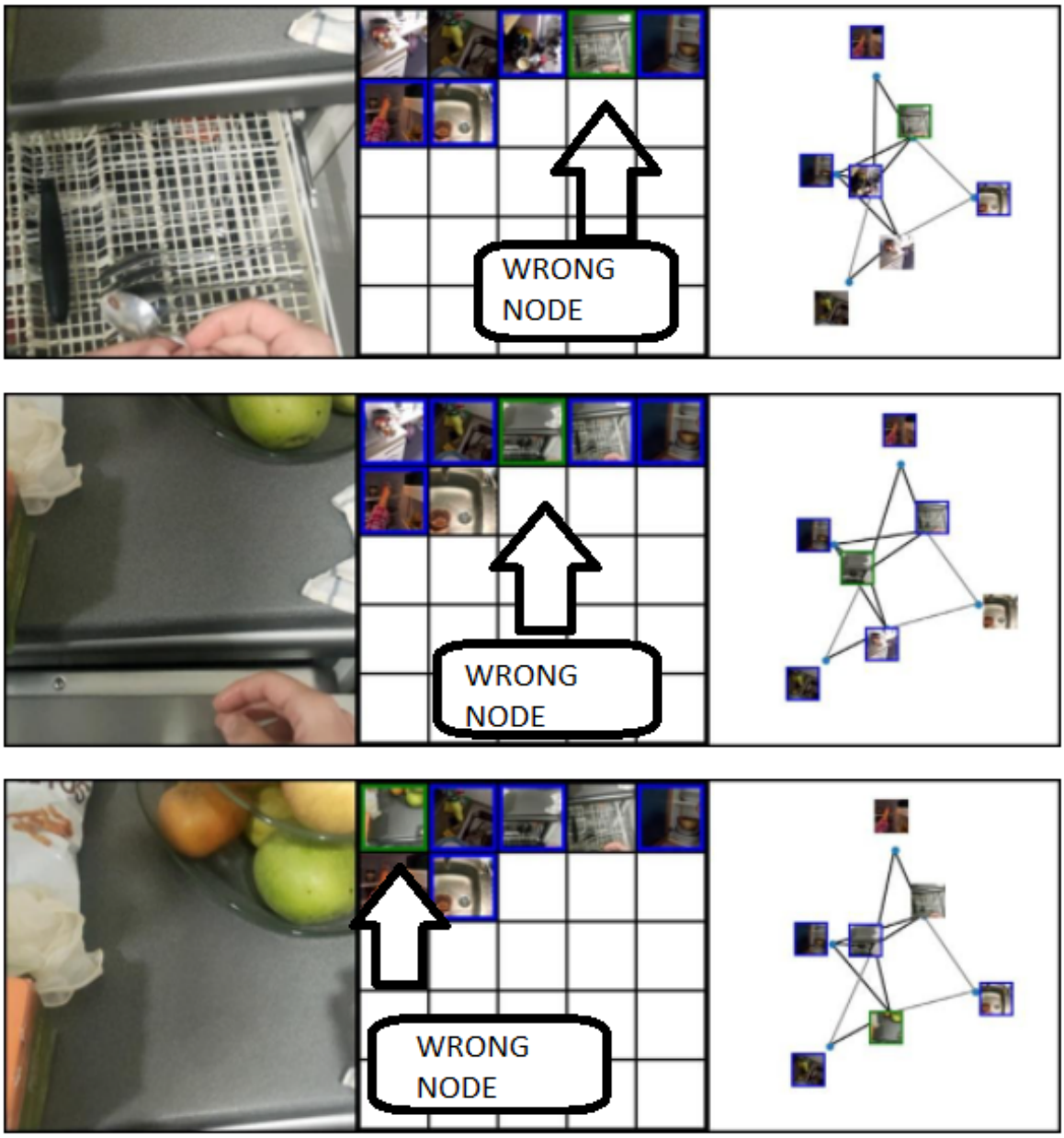

Figure 3.18: Final metric on second limitation

K visits with the lower value were selected.

The option chosen limits the number of visits and helps the system to have more heterogeneous visits. This way, the same region with changes can easily be associated. This, together with the new comparison method will help to limit the computing time and maybe reach at least similar results. Overall, the new method is at least better than a random one, and allows the system to have similar results with the use of fewer visits.

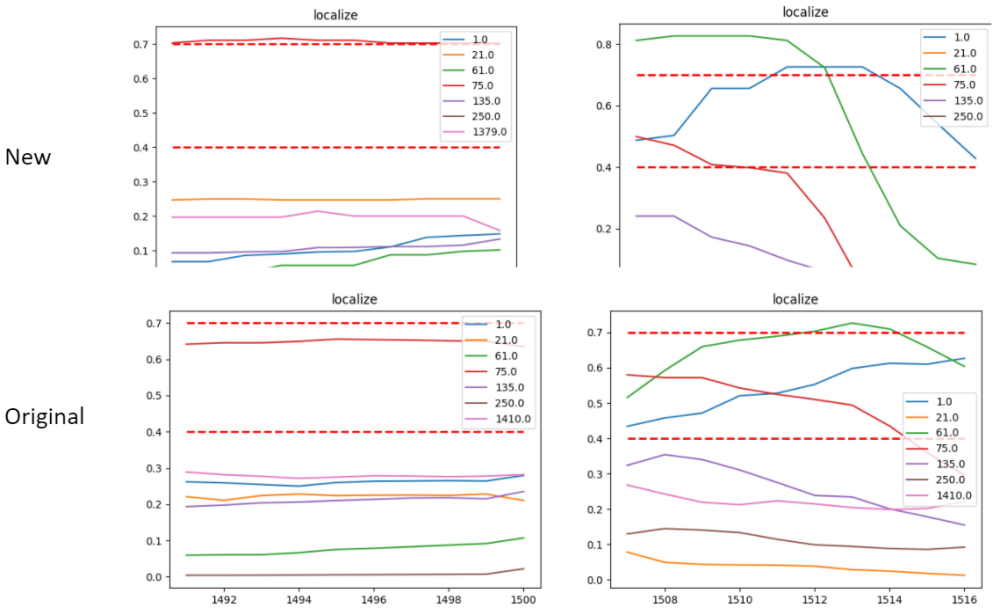

Figure 3.19: Specific scores with the final metric on external video

Chapter 4

Experimental evaluation

4.1 Chapter overview

This chapter evaluates the benefits of the changes described in the previous one using a larger set of videos. It also presents the application.

With the exploration of some videos we realize that results of the new method are very similar to those of the baseline method: the amount of nodes is more or less the same (sometimes less with the *new approach*) but the main difference lies in how the method associates visits to nodes, and when it instead creates a new node. This is logical because the changes previously done are in this line.

To recap, there are two principal differences in the operation introduced by the changes:

- Sometimes the *new approach* detects the previous node and it does not create a new node or it does not get stuck.
- Other times the *baseline approach* creates more than one node for the same area, and with the *new approach* the same number of nodes are created but they are generated in a more coherent way. E.g., in the sink area with the *baseline approach* the algorithm creates two nodes with identical appearance, but with the new one creates one node for the part of the sink where the dishes are, and the other one for the faucet.

Previous changes explanation. Both changes are explained by the new improved association method.

- **First change**. This is explained in the *first limitations solution* part (Section 3.6), and it was the problem solved.
- **Second change**. The new method system associates more precisely with existing areas but when a new part with different features appears, it is unavoidable to create a new node. With the *baseline approach* when a new region part appears, the system creates a new node that becomes the main one for that region (as it associate worse with previous nodes). Using the *new approach*, if a new part of the region appears it creates a new node but just for this part (the other part is associated with previous node as the association in this system is stronger), so now the region is divided into different nodes representing different parts of the region.

First change have improved the operation of the system at some point. With the second change we can discuss if this is a positive modification or not, it has advantages and disadvantages. For instance, it has more sense to divide the region in these more or less different parts, but it produces the system to has more transitions between nodes in a same region which can derive in additional problems:

- More visits means more computational cost, this was in part solved by the new limitation on visits, but it reaches the 20 visits limit faster than the baseline, therefore the computational cost increases anyway.

- A higher number of transitions between nodes may affect the metrics as we are going to see later.

Previous results are subjective, there are positive points and negative points, and obviously depend on the video. The next section deals with a more objective way to evaluate which method is better and analyze their strong and weak points.

4.2 A methodology for generating places annotations.

Our aim is to get an objective way to evaluate the different methods. To achieve this goal ground-truth annotations are required, in this case, region annotations. However, the annotations in the epic-kitchens database are for actions. As explained in the Chapter 2 and in Section 3.2.2, the network is trained by detecting the places based on human actions. Almost all the places detected by the system are regions where an action is performed as we can observe in figure 4.1. We can use this fact to create a semi-automatic way of annotating the video with region labels. With this idea we reduce the time needed to create the region's ground-truth, because in the other way, we need to go frame by frame and video's length goes from 20 000 to more than 100 000 frames each one.

The idea is the following:

1. Places detected grouped in nodes are obtained with the algorithm, this way we have the frames grouped and the work is simplified.

2. Now the automatic region classification is refined with the Epic-Kitchens actions, following a protocol explained below.

3. Finally, with these refined groups we manually annotate the specific region. In this manner we avoid the problem of dividing a same region in different nodes, which is the main issue of the system and we achieve the semi-automatic approach.

With this idea we only need to define these groups of frames, and the work is greatly reduced.

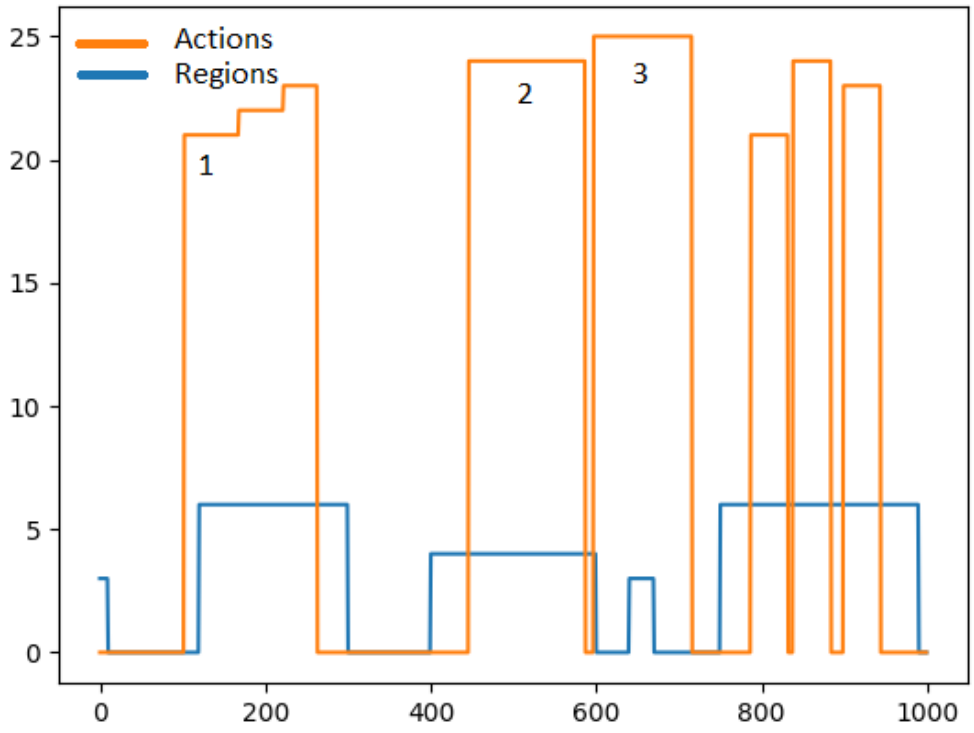

Figure 4.1: Action annotations and regions detected

4.2.1 Protocol

- **Action and region partially overlap.** In this case we consider two cases:
  - **Most of the action is in the region.** If this is the case we increase the region frames to include the entire action. Here we need to consider the actions which could be present in different nodes, in this case, we can not move the edge of the region to the edge of the action because this way we are overlapping regions, and it is not correct for two regions to overlap frames. If this happens we move the edge of the current regions to the edge of the region sharing action. This is the case *1* in the Figure 4.1.
  - **Most of the action is not in the region.** Here we move the edge of the region to not include the action. There is not a graphic example in the figure but it is easily understandable with the previous one.
- **Region totally included in an action.** This is similar to the previous case but on both sides. We move both edges to include the entire action (Case *2* in the Figure 4.1). Here we have a similar problem to the one explained in *Most of the action is in the region.* There could be more regions on the action, and if we do this, all regions would include the frames and overlap all of them. In this case we only move the edges of the region to the edges of the regions around. This particular case almost never occurs, but we need to consider it.
- **Action totally included in a region.** Here we do not change anything, because

an action can start and end inside a region. Case *3* in the Figure 4.1.

We can see the changes on the original regions detected by comparing Figure 4.1 an Figure 4.2. With these new refined groups the third step is carried out, in which we simply manually name the regions (e.g., we change 1 to sink).

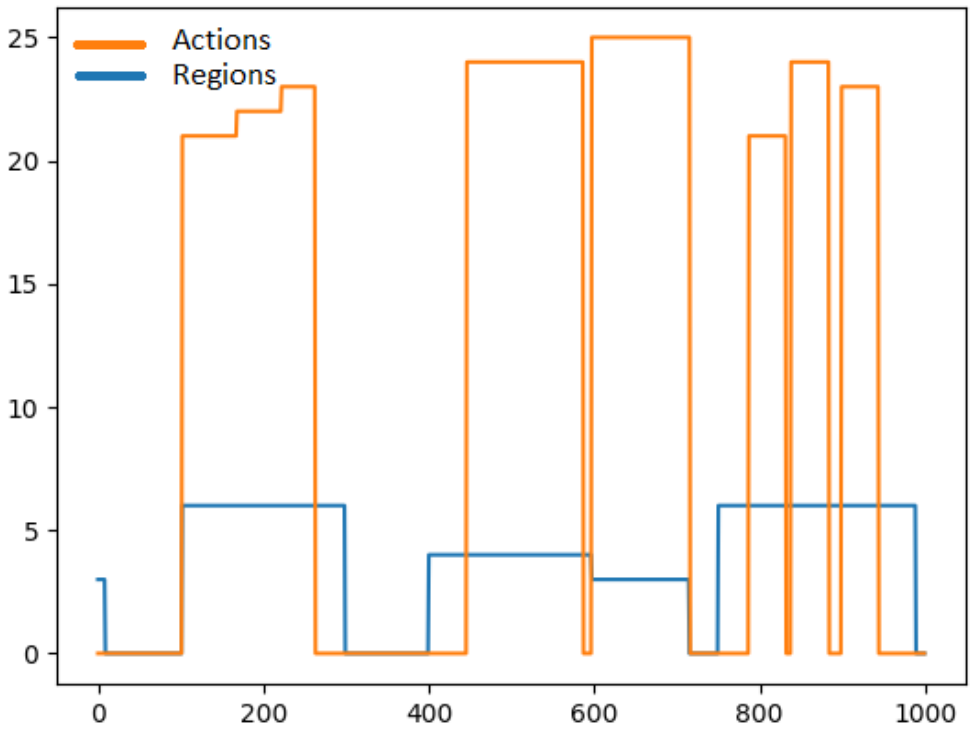

Figure 4.2: Action annotations and regions detected after tuning

4.2.2 Handling unfairness in evaluation.

The annotation process needs an initial approach to obtain the initial node partitions. If we are going to evaluate the two methods, we need to choose one of them to address the first point. A possibility is to use the *baseline approach*, because it is a little bit less expensive in computational terms. But as the annotations are a modification of the method output, the method from which the annotations come could be benefited, yielding a slightly unfair comparison.

Solution implemented. To avoid this problem, we are going to create the annotations with both methods as an initial point. The evaluation will consist of obtaining the metrics for each method with both annotations, this way we balance the method biases.

Other possible solution. The previous one is not the only solution, there are other possibilities. Another option thought was the following:

1. Run the algorithm with one of the approaches.
2. Refine the results with the action annotations.
3. With these results, use our new functionality to create a graph starting from a previous one and as previous one use the one refined (previous point). The results of the association to the previous nodes is the graph we are going to use until now. With this graph go back to point 2 and repeat it T times.
4. Finally after the T executions the manual annotations are done.

With this idea the results should be more generic and work similar for both approaches, but we finally dismissed it because of the computational cost it supposes. The initial execution with the shortest videos is approximately 2 hours, but the combined graph is of roughly 5 hours, and we need to repeat this T times. With longer videos combined graphs can go up to 15 hours easily.

Eventually we decided to create the annotations with the first idea. The annotations were done for the user P22 of the Epic-Kitchens database, we have selected this user because is the one with a largest quantity of action annotations.

4.3 Performance metrics

Once the annotations are computed we can evaluate the results and obtain an objective comparison between both methods. But now we need to define appropriate metrics to use the annotations.

There are several points that need to be evaluated, the overlap of regions and not regions parts (that we called transition regions), and the *dispersion* (amount of nodes on each region). The metrics can be separated in two types: frame-level, and region-level metrics.

- **Frame-level metrics**. These metrics evaluate the results at frame level:
  - **Dispersion metric**. To measure the *dispersion*, we defined a metric; in a manually defined region, for example the fridge, the *dispersion* is defined as the percentage of the node with the biggest occupation divided by the number of nodes that this region has. This metric favours the regions divided in less nodes, and also larger occupation of the predominant node.
  - **EOA metrics**: to compute the *recall*, *precision*, *accuracy* and *F1 score*, we need to define what are true positives, true negatives, false positives, and false negatives in our system. To clearly show this we have the figure 4.3. These metrics are used to measure the overlap of region and not region parts.
- **Region-level metric**. Frame level metrics are affected by little deviations in the method results. Sometimes defining where a region begins and ends is complicated and extremely subjective, and deviated predictions do not mean that a method is worse than another, therefore a region metric was also used.
  - **Accuracy Accumulated Area (AAA)**: This is the area under a curve (but not the usual AUC metric). The curve is composed of 11 values; each

value is obtained from a threshold swept from 1.0 down to 0.0 using 0.1 steps. What we do is to get for each annotation the percentage occupied by each method's result, and if it is over a given the threshold we sum up a 1 (region detected) and if not a 0 (region not detected). Then, we compute the mean of every piece. So, finally, we get a value for each threshold starting always from 0 (because it is impossible to yield a 100% of the annotation occupied) and ending with a 1 (it is impossible to not be over a 0% of the piece occupied). The faster the curve grows, the higher the AAA is. The Figure 4.4 shows an example of this curve.

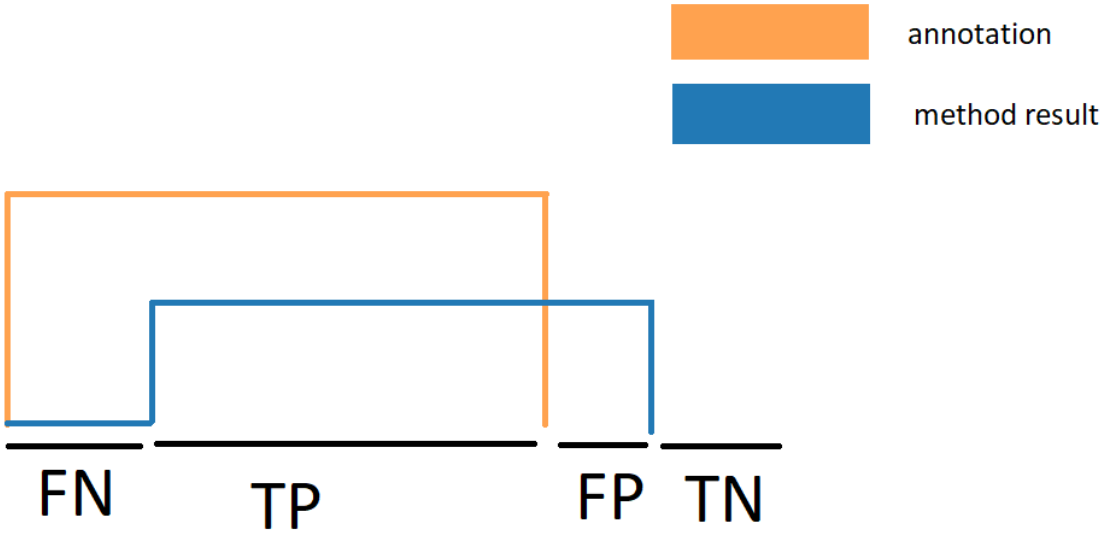

Figure 4.3: Matrix confusion values explained

With these metrics we evaluate the majority of the modifications performed on the system, and we can get reliable values. Every metric would be better or worse depending on several aspects, for example as the second method has a bigger amount of transitions, the overlap with region parts should be worse, but since it spends less time in the transition zone (remember that this was the problem we solved in the previous chapter with the sink), it maybe makes up for the above fact. We will explore these results in the following section.

4.4 Experimental results

Now we can use all the work done to finally get an objective way to evaluate both methods. The procedure is the following:

1. As both methods have 2 parameters, we are going to tune both of them to get the best values for each method. The original algorithm has the lower threshold at 0.4 and the upper threshold at 0.7. Due to the high computational cost we can not do a big tuning, so in the lower threshold we are taking [0.3 0.4 0.5] and in the upper threshold [0.6 0.7 0.8]. We carry out the tuning with every possible combination of both thresholds (9 combinations).

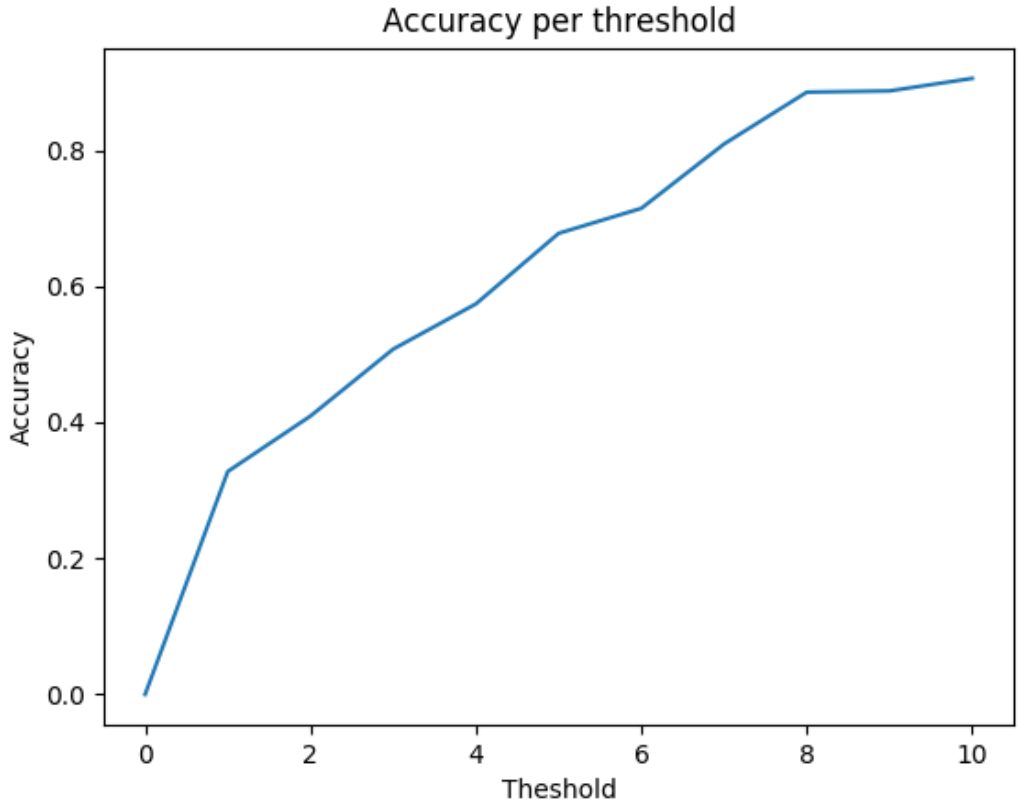

Figure 4.4: Accuracy Accumulated Area (AAA)

2. With the previous tuning we have 9 executions per video per method, at first we wanted to use the entire user P22, as we have calculated the annotations for it (we have selected it because it is the more complete in terms of action annotations). But again, the computational cost limited us. To run the algorithm for the whole user (17 videos) with 18 executions on each one (9 on each method), the time required would be months, so we decided to use only three videos. The videos were chosen for their length (not the longer ones) and characteristics (videos where the different places appear multiple times). This took us approximately 2 weeks.

4.4.1 Methods evaluation

With the videos computed, we finally got the metrics. As the F1 score depends on the precision and recall, we are not going to use it in the evaluation. Therefore we have 5 values per video execution, and there are 9 executions per video, that means we have a 9×5 table per video. We have two methods so there are two of these tables. As we said in Section 4.2.2 to avoid possible inequalities due to the initial method used, we are going to evaluate each method using annotations refining initial predictions of both methods, therefore we really have 4 tables to evaluate each video.

The individual results on a video do not matter to us, we want to globally compare the methods. To achieve this we are going to average the corresponding performance metrics on different videos. This way we finally got 4 tables to evaluate our two method,

Table 4.1: New method with new method annotations

Thresholds	(0.3,0.6)	(0.3,0.7)	(0.3,0.8)	(0.4,0.6)	(0.4,0.7)	(0.4,0.8)	(0.5,0.6)	(0.5,0.7)	(0.5,0.8)
AAA	0.81	0.80	0.75	0.79	0.76	0.69	0.80	0.80	0.70
RECALL	0.94	0.93	0.89	0.94	0.94	0.88	0.93	0.93	0.89
PRECISION	0.56	0.60	0.64	0.66	0.68	0.70	0.74	0.74	0.72
ACCURACY	0.59	0.62	0.64	0.68	0.69	0.68	0.74	0.75	0.71
DISPERSION	0.35	0.27	0.24	0.33	0.31	0.28	0.32	0.30	0.28

Table 4.2: New method with original method annotations

Thresholds	(0.3,0.6)	(0.3,0.7)	(0.3,0.8)	(0.4,0.6)	(0.4,0.7)	(0.4,0.8)	(0.5,0.6)	(0.5,0.7)	(0.5,0.8)
AAA	0.74	0.78	0.70	0.78	0.74	0.68	0.78	0.81	0.70
RECALL	0.92	0.92	0.88	0.95	0.93	0.90	0.93	0.92	0.89
PRECISION	0.54	0.61	0.60	0.61	0.67	0.67	0.66	0.71	0.69
ACCURACY	0.60	0.67	0.63	0.67	0.75	0.71	0.72	0.78	0.74
DISPERSION	0.28	0.26	0.28	0.32	0.32	0.31	0.30	0.30	0.31

Table 4.3: Original method with original method annotations

Thresholds	(0.3,0.6)	(0.3,0.7)	(0.3,0.8)	(0.4,0.6)	(0.4,0.7)	(0.4,0.8)	(0.5,0.6)	(0.5,0.7)	(0.5,0.8)
AAA	0.79	0.74	0.67	0.83	0.76	0.66	0.77	0.75	0.62
RECALL	0.92	0.92	0.87	0.94	0.92	0.87	0.90	0.90	0.84
PRECISION	0.60	0.65	0.67	0.66	0.64	0.65	0.73	0.73	0.75
ACCURACY	0.65	0.72	0.72	0.71	0.71	0.72	0.77	0.78	0.78
DISPERSION	0.31	0.26	0.25	0.39	0.33	0.3	0.36	0.30	0.29

Table 4.4: Original method with new method annotations

Thresholds	(0.3,0.6)	(0.3,0.7)	(0.3,0.8)	(0.4,0.6)	(0.4,0.7)	(0.4,0.8)	(0.5,0.6)	(0.5,0.7)	(0.5,0.8)
AAA	0.75	0.73	0.68	0.78	0.76	0.69	0.75	0.72	0.65
RECALL	0.93	0.90	0.87	0.92	0.91	0.89	0.88	0.88	0.82
PRECISION	0.62	0.63	0.66	0.62	0.64	0.65	0.69	0.70	0.69
ACCURACY	0.64	0.63	0.65	0.64	0.65	0.66	0.68	0.69	0.71
DISPERSION	0.29	0.20	0.19	0.31	0.26	0.25	0.26	0.27	0.20

which make the task more accessible. It would have been more reliable to use more videos, so that results would have been more general, but as we have explained, this was not possible.

4.4.2 Final decision about the parameters and method based on the evaluation results

For the different thresholds the evaluation values follow a pattern:

- **AAA**: With a lower *upper threshold* the results are better. For the *lower thresholds* results are more or less similar. The reason is that the system associates easier, so it has more frames overlapping and it is easier to overcome the threshold given for this metric.

- **RECALL:**
$$RECALL = \frac{TP}{TP + FN}$$
Again with a smaller *upper threshold* the results are greater. Here for the *lower threshold* the better option is the middle one. The recall is favored by an easier association, because this metric only takes into account the frames inside the place annotations (TP and FN Figure 4.3), therefore with more frames associated (lower *upper threshold*) better results.

- **PRECISION:**
$$PRECISION = \frac{TP}{TP + FP}$$
The general pattern here is contrary to the previous ones, with a higher *upper threshold* the results are improved. For the *lower threshold*, the best option is the highest one. The precision takes into account the overlap with annotations and detection (TP and FP Figure 4.3); if the system detects more frames but they are wrong this metric suffers. If the system create less nodes the wrong frames are reduced and the metric goes up, for this reason the highest *lower threshold* give the best results, but this is not an indicator of a better performance, but of a *less bad* performance.

- **ACCURACY:**
$$ACCURACY = \frac{TP + TN}{TP + TN + FP + FN}$$
The accuracy takes into account in some way both **RECALL** and **PRECISION** metrics, because it computes the errors in and out the annotations. For this reason, it is more balanced but better for the intermediate *upper thresholds*. For the *lower threshold* the results are a little bit better with the highest one than with the intermediate one. This is due to the highest threshold in the *precision metric* is more favored than the intermediate one in the *recall*. The benefit suffered by the highest threshold in the *precision metric* is as we explained because it is *least bad* with fewer nodes but not for a proper performance.

- **DISPERSION**: Again, with a lower upper threshold, this metric works better. This way the system gives a higher occupation to the corresponding node of each region. The lower threshold is more beneficial with an intermediate one, not being either permissive or restrictive.

These facts are present in the four tables 4.1 4.2 4.3 4.4.

With the previous results we can make a decision:

- **Upper threshold**: Since both smaller and bigger *upper thresholds* have benefits and issues, the one which is favored is the middle one. We saw this fact in the accuracy which is a mix of the recall and precision. And this way, some aspects of the algorithm are not impaired with respect to others.

- **Lower threshold**: The *AAA* is not really affected by this threshold. *Recall* works better with an intermediate threshold and *precision* with a higher one, but as we have explained on the *accuracy*, the benefit in precision comes from being is the *least bad*. At last, dispersion prefers an intermediate threshold. Metrics indicate directly that the best is the intermediate one.

With this in mind, the final threshold decision is the pair (0.4, 0.7) i.e. the original ones. It makes sense because *Ego-Topo* creators have verified for sure that these are the most appropriate ones, with a much bigger tuning.

Now we can create a table 4.5 with the chosen values for each evaluation to finally get a comparison between both methods. Just looking at this table, we realize that the new method works better using either annotations refined from it results but also using those refined from the original method results. Overall, the new system has an improved performance.

Table 4.5: Comparison between methods

Method/annotations	New/Original	New/New	Original/Original	Original/New
AAA	0.74	0.76	0.76	0.76
RECALL	0.93	0.94	0.92	0.91
PRECISION	0.67	0.68	0.64	0.64
ACCURACY	0.75	0.69	0.71	0.65
DISPERSION	0.32	0.31	0.33	0.26

4.5 Discussion

We finally got an objective evaluation that tells us which is the better system, we no longer depend on subjective evaluations. Of course, both approaches have weak and strong points, but the general conclusion is that the new method is slightly superior to the previous one.

We achieve this improvement without re-training the net, the localization network was trained by Facebook, and for sure is really accurate (At least for videos similar to those used for training it). But on the other side, the algorithm has some points for improvement. Sometimes to reach a more promising performance, the weak point is not the network but how the network is used. The enhancement is not exaggerated, but it fixes some remarkable issues that could have caused us problems to reach a good performance application.

4.6 Software demonstration

Finally, a *new approach* has been achieved that solves the problems of the beginning. In addition, its use has been justified thanks to the evaluation. With this enhanced approach, now we are going to develop the application. The final purpose is to get statistics on whether two users have been in the same regions or not. If they have shared regions we could denominate this as an direct or indirect contact, depending on sharing region simultaneously or one after the other, and the necessary measures could be taken.

The new combined graphs functionality allows us to know if two users have been in the same region. A visit added to a previous node means that both users have passed by there. If we indicate the exact time at which each video begins, we can also

determine if they have matched the regions simultaneously or not. This is more or less what the application will do with some changes:

1. We get an initial graph with the regions presented in the space to be analyzed.
2. With this initial graph we got a combined graph for each user video.
3. We are going to assume that both users enter the region at the same time, and we will get if they have been in the same regions, and if they have been in them at the same time, showing in that case, how long they have remained together.

The reason we have done it this way is that it is more secure if three detections match than if only two match. In the figure 4.5 we can see an example.

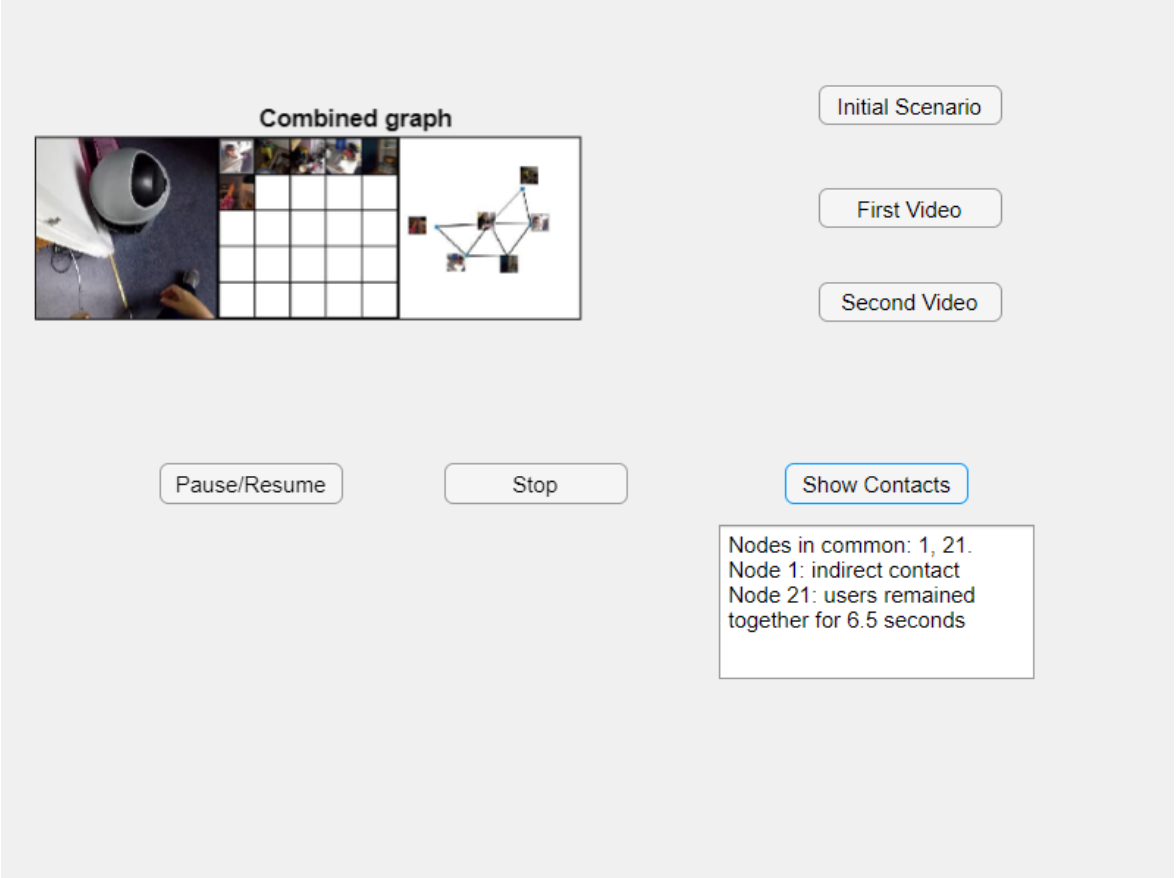

Figure 4.5: Application example

Chapter 5

Conclusions and future work

5.1 Conclusions

Carrying out this work has been a very enriching task, the knowledge acquired covers many fields, and the conclusions are abundant.

- Automatic place registration is quite a difficult task due to the nature of a place. Places are locations where a user carries out different tasks and the visual features of the place could change over time. Therefore, to detect a place we need to understand its purpose, and in the approach identify the place by the use it has for the user.
- Current state-of-the-art place registration method is *Ego-topo*, and despite the performance is surprising, it can be improved. The system is amendable especially in algorithm parts that do not have to do with the network. **To improve a Deep Learning system, sometimes you do not need to improve the network itself**, particularly in networks trained by large corporations where it is usually the strong point of the system.
- Computational cost could be the biggest bottleneck for a given task. In lightweight tasks we do not realize, but when you come across a heavy computational cost task, you understand the relevance of doing things as efficiently as possible.
- In video occupations like this, the hard disk space needed could make unfeasible the use of a big amount of videos in a domestic machine without the use of an external tool.
- Having a register of the places that a person has visited is a very powerful tool. It can be used from tracing COVID-19 contacts to model routines of a person to teach a machine and many more applications.
- There are a lot of transverse conclusions acquired during the accomplishment of this work, here we have some:
  - The importance of using graphics to analyze results.
  - The big number of problems that appear when you do a work like this.
  - The complexity of installing an external code if the instructions are not clear enough.

– The need to automate as much as possible to avoid errors and speed up work.

5.2 Future work

Future work is a must, because the current system is far from perfection, there are a lot of ways to improve the actual results:

1. **Algorithm aspects**: As we change the metric and the visits selection there are other aspects in the code that could be improved, but the lack of time did not allow us to do it:

   - *Better visit frames selection*: as we have explained, the system uses only central frames of each visit, but a smarter frame selection of the most representative frames of the visits could improve results.
   - *Blurring elimination*: Videos have blurring frames, and these frames could be associated to a region and used for future comparisons, adding noise to the results. This fact is partially solved by the new method, but a detection and elimination of these frames can be a promising option.
   - *Improve our method*: The changes we have made do not have to be the immutable option, maybe another association/creation metric is stronger, or there is a more appropriate way to select the node visits.

2. **Network aspects**: Other options could be to improve the network by deep learning techniques (such as fine-tuning, continual learning, etc.) or creating a new network with other methods (different architecture, other learning technique, etc).

3. **Use other information**: Here we only use a RGB camera, but other information could be used to improve the system:

   - *depth information* to have more scene information and maybe better results.
   - *sound information*, this is a strong point. The sound information is in general really useful, but in this case quite more, every place has a different sound, for example the sink has water sound, the cut table knife sounds, etc. The mix of this information with the images could reach very promising results.

   Apart from this, the system could be adapted to operate on a bigger environment like an entire house, or an airport, where this would be much more useful.

Bibliography

[1] e. a. NAGARAJAN, Tushar, "Ego-topo: Environment affordances from egocentric video," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 163–172, 2020. 5, 6, 9

[2] e. a. DAMEN, Dima, "Scaling egocentric vision: The epic-kitchens dataset," *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 720–736, 2018. 5

[3] M. CAI, K. M. KITANI, and Y. SATO, "Understanding hand-object manipulation with grasp types and object attributes," *Robotics: Science and Systems.*, 2016. 5

[4] M. MA, H. FAN, and K. M. KITANI, "Going deeper into first-person activity recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1894–1903, 2016. 6

[5] Y. LI, M. LIU, and J. M. REHG, "In the eye of beholder: Joint learning of gaze and actions in first person video.," *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 619–635, 2018. 6

[6] e. a. PIRRI, Fiora, "Anticipation and next action forecasting in video: an end-to-end model with memory," *arXiv preprint arXiv:1901.03728*, 2019. 6, 7

[7] A. FURNARI and G. M. FARINELLA, "What would you expect? anticipating egocentric actions with rolling-unrolling lstms and modality attention," *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6252–6261, 2019. 6

[8] B. F. Y. Shi and R. Hartley., "Action anticipation with rbf kernelized feature mapping rnn," *ECCV*, 2018. 6

[9] X. WANG and A. GUPTA, "Videos as space-time region graphs," *Proceedings of the European conference on computer vision (ECCV)*, pp. 399–417, 2018. 7

[10] e. a. GIRDHAR, Rohit, "Actionvlad: Learning spatio-temporal aggregation for action classification," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 971–980, 2017. 7

[11] e. a. WU, Chao-Yuan, "Long-term feature banks for detailed video understanding," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 284–293, 2019. 7

[12] e. a. ALAYRAC, Jean-Baptiste, "Joint discovery of object states and manipulation actions," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2127–2136, 2017. 7

[13] H. S. KOPPULA and A. SAXENA, "Physically grounded spatio-temporal object affordances," *European Conference on Computer Vision*, pp. 831–847, 2014. 7

[14] e. a. DELAITRE, Vincent, "Scene semantics from long-term observation of people.," *European conference on computer vision*, pp. 284–298, 2012. 7

[15] N. RHINEHART and K. M. KITANI, "Learning action maps of large environments via first-person vision," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–588, 2016. 8

[16] e. a. KOILE, Kimberle, "Activity zones for context-aware computing," *International Conference on Ubiquitous Computing*, pp. 90–106, 2003. 8

[17] A. D. N. Savinov and V. Koltun, "Semi-parametric topological memory for navigation," *ICLR*, 2018. 10